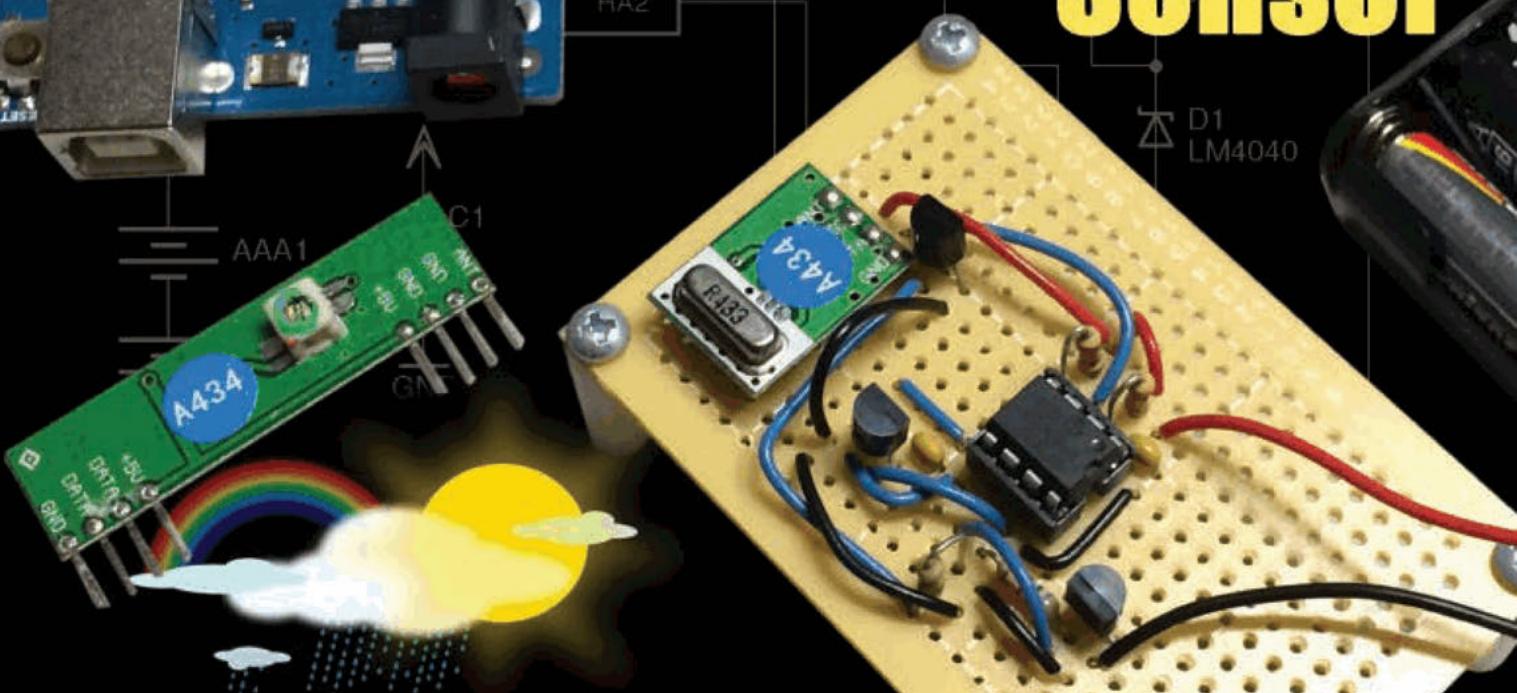




NUTS AND VOLTS

EVERYTHING FOR ELECTRONICS

Easy-to-Build PIC-Based Remote Temperature Sensor



- ♦ **Clocks For PICs**
Everything you need to know!
- ♦ **The Arduino Classroom**

Now In Our
35th YEAR!

Serial-to-Ethernet Solutions with SSL & SSH Security

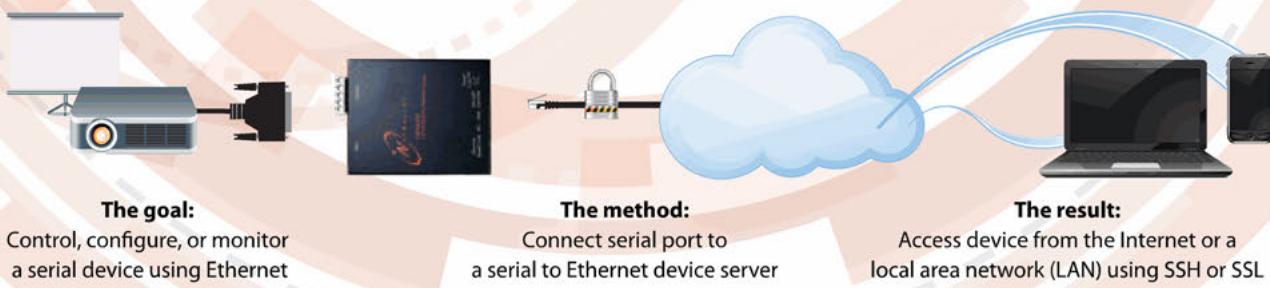
➤ **SB70 LC**
2-Port 3.3V TTL serial device support

➤ **SB700 EX**
2-Port RS-232 & RS-422/485 serial device support

➤ **CB34 EX**
2-Port RS-232 & RS-422/485 serial device support

➤ **PK70 EX 232/485/MMS**
4-Port RS-232 and/or RS-485/422 serial device support

Access serial ports over SSL & SSH protected Ethernet!



NetBurners Secure Serial to Ethernet Servers network-enable serial devices out of the box - no programming or development is required. The hardware is pre-programmed to convert your serial data to Ethernet, enabling secure communication with the serial device over a network or the Internet.

Part No. SB70LC-100IR.....	\$49 (qty. 100)
Part No. SB700EX-100IR.....	\$149 (qty. 100)
Part No. CB34-EX-100IR.....	\$179 (qty. 100)
Part No. PK70EX-232CR.....	\$285 (qty. 100)
Part No. PK70EX-485CR.....	\$288 (qty. 100)
Part No. PK70EX-MMSCR.....	\$346 (qty. 100)

Need a custom solution?

NetBurner Development Kits are available to customize any aspect of operation including web pages, data filtering, or custom network applications.

➤ **For additional information please visit**
<http://www.netburner.com/kits>



NEW YEAR'S RESOLUTION

- ✓ Cut down on alcohol and chocolate
- ✓ Go to gym twice a week
- ✓ Use only lead-free soldering paste
- ✓ Replace EasyPIC6 with EasyPIC7
- ✓ Try 32-bit microcontrollers
- ✓ Start saving money

Use voucher code **HAPPY2014**
during checkout to **get 10% discount**
on all orders **until January 31st 2014**

January 2014

28 Build a PIC-Based Remote Temperature Sensor

You can never have too many remote sensors, right? This circuit lets you set up a homemade weather station to monitor the most popular of all parameters: temperature.

■ By Ryan Clarke



34 The (Nearly) Universal Process Monitor

While the circuit described here is specifically for showing system parameters in an electric truck conversion, you can utilize it as a process monitor and display for analog and pulsed signals in a variety of applications.

■ By Daniel Gravatt

Columns

08 TechKnowledgey 2014

Events, Advances, and News

Read about a transistor that learns, secure wireless access, the Smellophone, plus some other cool stuff.

12 The Spin Zone

Adventures in Propeller Programming

Shifty Business.

Sometimes the simple tried-and-true solutions are overlooked because of the ever-increasing sophistication of microcontrollers.

Case in point: shift registers.

20 Q & A

Reader Questions Answered Here

Questions on LED drivers, remote control, and video and power over cable are answered.

52 Smiley's Workshop

Programming • Hardware • Projects

The Arduino Classroom.

Even though the Internet is flowing with tutorials on how to use the Arduino microcontroller, sometimes it's nice to have everything put together into an easy-to-follow introductory course. That's what we've done here.

38 All About Clocks for PICs

Taking the time to understand how clock options really work in a microcontroller will make implementing them that much easier.

■ By Thomas Henry

44 MakerPlot — The DIY Software Kit Part 4

Learn how to plot both analog and digital data from the Arduino Uno, then see how to log that data and play it back for greater analysis.

■ By John Gavlik and Martin Hebel

Departments

06	DEVELOPING PERSPECTIVES <i>Up Close & Personal</i>	65	ELECTRO-NET
07	READER FEEDBACK	74	NV WEBSTORE
24	NEW PRODUCTS	77	CLASSIFIEDS
27	SHOWCASE	78	TECH FORUM
		81	AD INDEX

62 Near Space

Approaching the Final Frontier

A New Airframe Design for Near Spacecraft — Part 2.

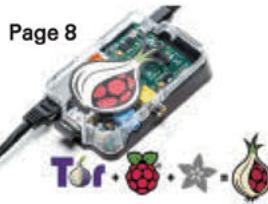
Building a new airframe is only part of creating a near spacecraft. There are things like post burst chaos that have to be considered.

67 The Design Cycle

Advanced Techniques for Design Engineers

Utilizing Embedded Web Development.

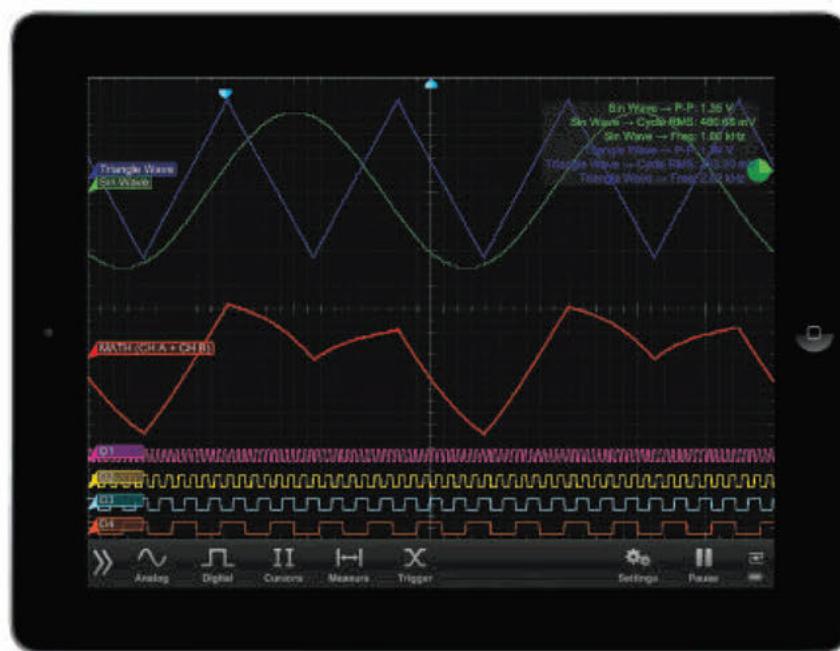
Remember drilling that slightly off-center toggle switch mounting hole for your project's front panel then have it be a bit too large? Those days are long gone. Trade in that drill and dull set of metal bits for some silicon versions that machine a perfect virtual front panel every time.



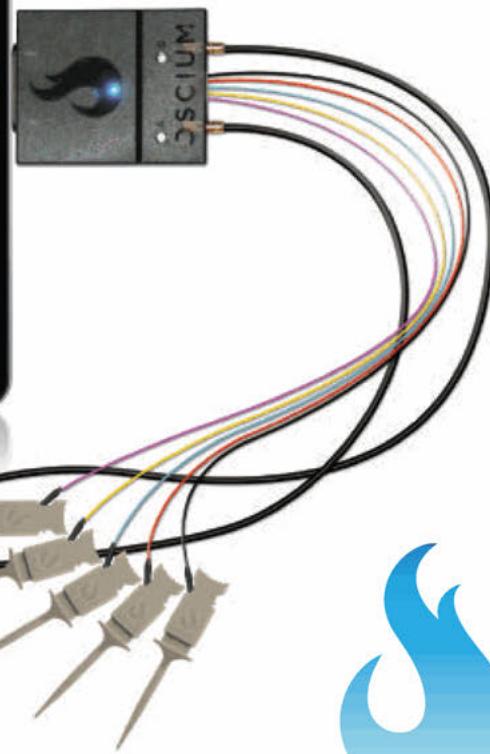
Page 52

Make The World Your Lab

Transform iPad, iPhone, & iPod Into An Oscilloscope



- Ultra Portable
- Two Analog Channels
- 4x Faster Than iMSO-104



OSCIUM

	iMSO-204	iMSO-104
Channels	2 Analog + 4 Digital	1 Analog + 4 Digital
Max Sample Rate	50 MS/s	12 MS/s
Bandwidth	5 MHz	5 MHz
Sample Depth	1000 pts	240 pts
Horizontal Sensitivity	200ns/div-10s/div	2μs/div-1s/div
iOS App	Download on the App Store	Download on the App Store



by Bryan Bergeron, Editor

DEVELOPING PERSPECTIVES

Up Close & Personal

As an electronics enthusiast with a bent towards construction, you can get by with a handful of tools. A good soldering iron, diagonal cutters, needlenose pliers, DMM, and a set of screwdrivers will get you through most projects. However, certain luxuries can make building and debugging hardware a joy. I'm talking about magnification tools, from the Luxo illuminated 3X magnification desk lamp to the palm-sized electronic magnifiers with 10x to 100x magnification that can enable you to read the values printed on SMT resistors.

I've owned one model or another of those Luxo magnifiers for most of my life. They're good for 'big picture' magnification. Large illuminated glass magnifiers provide a clear image of the area, good elbow room, and they're easy to work with. Because you're looking straight through a lens at the work area and your hands, there's

nothing to learn. What you see is what you get. Unfortunately, every time I check the prices, these magnifiers seem to increase in cost.

The situation is reversed with electronic magnifiers. Moreover, the feature-to-cost ratio continues to climb, with the latest models providing amazing results for relatively little outlay. At the high end of the electronic magnifier spectrum is the ProScope lineup. I've owned the original ProScope and the ProScope HR, in part because the lenses are compatible across the bodies — just like the lenses of an SLR camera. The latest models — ProScope HR2 and ProScope Mobile — provide enhanced resolution (1600 x 1200) and the ability to display and capture images on all Apple iOS devices. The latter could be especially useful if you're teaching an electronics class to a room full of iPhone and iPad users.

You can purchase the original ProScope base for \$99 (refurbished), or a new ProScope HR base for \$170. The HR2 and Mobile units (with lens) sell for \$350 and \$450, respectively. Additional lenses range from \$50 to \$250, depending on magnification and built-in illumination.

On the more affordable end of the spectrum is the Celestron Handheld Digital Microscope 2MP (\$45, Amazon). This capable magnifier will reveal the most minute details of an SMT solder joint, with magnifications from 10x to 150x and built-in LED illumination. The software application that accompanies the magnifier isn't quite as polished as the version that ships with the ProScope, but considering the price it's quite a bargain. The Windows version has several features not found on the Mac application, but this is a minor annoyance.

In addition to buying a new stand-alone electronic magnifier, you can repurpose that old microscope in your basement with a new digital image capture device. I've had great luck with the Celestron Digital Microscope Imager (\$35, Amazon). You simply remove the ocular eyepiece or tube from your microscope and drop in the 2 MP USB cylinder. The better your optics, the better the image, and magnification depends on your microscope's capabilities. I've had trouble with the software, so use Adobe Photoshop for image capture. Otherwise, for the price, you can't beat the capability. Before buying the imager, I was using an adapter for my Canon digital camera. The adapter kit alone cost \$150, and there is no software.

So, there's a solution out there for just about any budget. If you don't own a magnifier, treat yourself. The next time you're debugging a circuit board trace, you'll be glad you did. **NV**



CNC Technology in your Shop for Less
Visit us at: www.CNCShark.com
From \$2899 - \$3999 use code: NV1213A

Published Monthly By
T & L Publications, Inc.
 430 Princeland Ct.
 Corona, CA 92879-1300
(951) 371-8497
FAX (951) 371-3052
 Webstore orders only **1-800-783-4624**
www.nutsvolts.com

Subscriptions

Toll Free **1-877-525-2539**
 Outside US **1-818-487-4545**
 P.O. Box 15277
 North Hollywood, CA 91615

FOUNDER

Jack Lemieux

PUBLISHER

Larry Lemieux
publisher@nutsvolts.com

ASSOCIATE PUBLISHER

Robin Lemieux
robin@nutsvolts.com

EDITOR

Bryan Bergeron
techedit-nutsvolts@yahoo.com

CONTRIBUTING EDITORS

Jeff Eckert	Russ Kincaid
Joe Pardue	John Gavlik
Martin Hebel	Thomas Henry
Ryan Clarke	Dan Gravatt
Fred Eady	Jon McPhalen
Paul Verhage	

ADVERTISING SALES EXECUTIVE

Richard Lockhart
richard@nutsvolts.com

CIRCULATION DEPARTMENT

subscribe@nutsvolts.com

SHOW COORDINATOR

Audrey Lemieux

MARKETING COORDINATOR

WEBSTORE

Brian Kirkpatrick
sales@nutsvolts.com

WEB CONTENT

Michael Kaudze
website@nutsvolts.com

ADMINISTRATIVE ASSISTANT

Debbie Stauffacher

PRODUCTION

Sean Lemieux

Copyright © 2014 by T & L Publications, Inc.

All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: 430 Princeland Court, Corona, CA 92879.

Printed in the USA on SFI & FSC stock.



READER FEEDBACK

Delighted with HexBright

Great article about the Hexbright and its development. My wife and I have been using the Hexbrights that our son gave us last Christmas for 11 months now. We use them nearly every day and are very pleased. Neither of us has any need for reprogramming the units. I use mine in my work repairing things, and got a nice little surprise when working inside a PC that the Hexbright (at its lowest light setting) showed the cooling fan as nearly stopped.

The frequency of the duty-cycle for low illumination was a near match for the fan's RPM. "Cool," I thought. Even more cool was when I pointed the light skyward in a light rain (at night) and watched the light trails of the descending droplets.

Ray Bryan

A Match Maker

This letter is written in reference to the Reader Feedback, "Matching Tubes," by David Asselin which ran in the September 2013 edition.

I started Soundmaster, Ltd., in Canada in the late 1940s, and the company was operational until my retirement in 2010. Over the years, we manufactured over 30,000 audio (tube) amplifiers in 29 active models and countless custom "specials."

The early amplifiers used 807 tubes which I acquired from war surplus. After we ran out of them, I modified my amplifiers to run 5881s. All output tube pairs were carefully matched on a device that I designed and built specifically for that purpose.

Basically, it was a normal output stage wired class "A" using cathode

Continued on page 50



EARN MORE MONEY
Get your dream job!

Be an FCC Licensed Wireless Technician!

Make up to \$100,000 a year and more with NO college degree

Learn Wireless Communications and get your "FCC Commercial License" with our proven Home-Study Course!

- No need to quit your job or go to school.
- This course is easy, fast and low cost.
- No previous experience needed!
- Learn at home in your spare time!

Move to the front of the employment line in Radio-TV, Communications, Avionics, Radar, Maritime and more... even start your own business!

Call now for FREE info: 800-877-8433
ext. 109

Or, email us:
info@LicenseTraining.com

COMMAND PRODUCTIONS, Warren Weagant's FCC License Training
 Industrial Center, 480 Gate Five Rd., PO Box 3000, Sausalito, CA 94966-3000



Post comments on this article at www.nutsvolts.com/index.php?/magazine/article/january2014_TechKnow14.

ADVANCED TECHNOLOGY

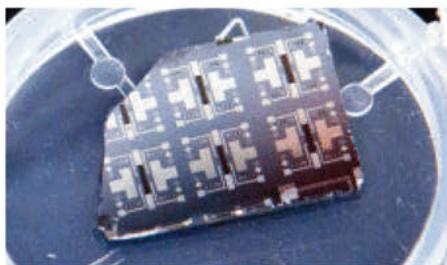


Photo courtesy of Eliza Grinnell, SEAS Communications.

■ Several prototypes of the synaptic transistor are visible on this silicon chip.

A Transistor That Learns

Ever since John McCarthy coined the term "artificial intelligence" back in 1955, computer scientists have been trying to find ways of implementing it — both through hardware and software — often while arguing about what the term actually means. Creating an intelligent machine that mimics the human brain is definitely a tall order. According to the University of Alberta's Prof. Chris Westbury, a typical brain has somewhere in the neighborhood of 100 billion neurons — each of which connects to about 1,000 other neurons — and all of them can fire about 200 times per second. This gives us an equivalent clock rate of 100 billion \times 200 \times 1,000 = about 20 million GHz. And it uses only about 20W of power to do it.

One of the main advantages the brain has over traditional circuitry is that the more times a synapse fires, the stronger its connections become. In a word, it learns. Transistors don't. At least they didn't until back in November, some materials scientists at the Harvard School of Engineering and Applied Sciences (www.seas.harvard.edu) created a new type of device that simultaneously modulates the flow of information in a circuit and physically adapts to changing signals. As described in a recent issue of *Nature Communications*, their synaptic transistor "could mark the beginning of a new kind of artificial intelligence: One embedded not in smart algorithms but in the very architecture of a computer."

According to co-lead author Jian Shi, "Each time a neuron initiates an action and another neuron reacts, the synapse between them increases the strength of its connection. And the faster the neurons spike each time, the stronger the synaptic connection. Essentially, it memorizes the action between the neurons."

The details are complicated, but basically a biological synapse employs calcium ions and receptors to learn, whereas the new device does the same thing with oxygen ions. As a result, it has a practically unlimited number of possible states rather than just "on" and "off." The challenge, as always, is to put the concept to practical use, but as the authors assure us, "This kind of proof-of-concept demonstration carries [our] work into the 'applied' world where you can really translate these exotic electronic properties into compelling, state-of-the-art devices." ▲

Dolphins Inspire New Radar

One of the many clever things that a pod of dolphins (and some whales) can do is blow bubbles. That isn't so clever in itself, but they cooperatively use blasts of seltzer-size bubbles to create a "net" to trap a school of fish so they can more easily scoop them up for lunch. A few years ago, Prof. Tim Leighton from the Institute of Sound and Vibration Research at the University of Southampton (www.southampton.ac.uk) reasoned that the dolphin's sonar must include the ability to distinguish between targets and clutter (i.e., fish and bubbles) in the turbulent water. This led to the concept of twin inverted pulse sonar (TWIPS) in which two pulses — identical except for one being phase inverted — are sent out in rapid succession. It turns out that the TWIPS signal enhances linear scatter from the desired target but suppresses nonlinear scatter from the bubbles, allowing easy target detection.

More recently, Prof. Leighton — no doubt after slapping himself on the forehead — realized that the principle should be applicable to electromagnetic waves in the form of twin inverted pulse radar (TWIPR). So, he and some cohorts put it to the test using a dipole antenna with a small (6 cm) diode across its feedpoint as the target, and a 34 x 40 cm aluminum plate and a rusty bench clamp as the clutter. Lo and behold, the target showed up 100,000 times more powerfully than the clutter. This is of significance — as the antenna/diode combination is typical of devices associated with covert communications, espionage, and explosives — so TWIPR instruments could come in handy for detecting such hidden devices. In addition, because the target weighs only 2.8 g, costs less than one Euro, and requires no batteries, they could be used in inexpensive location and identification tags for animals, infrastructure (pipelines, conduits for example), and humans entering hazardous areas — particularly where they might be underground or buried.

Professor Leighton added, "In addition to the applications discussed above, such technology could be extended to other radiations such as magnetic resonance imaging (MRI) and light detection and ranging (LIDAR), which, for example, scatters nonlinearly from combustion products, offering the possibility of early fire detection systems." All because dolphins blow bubbles. ▲



■ Bubble-blowing dolphins inspire a new type of radar.

COMPUTERS and NETWORKING

A Tabtop? A Laplet?

It appears that the marketing geniuses in the computer industry are convinced that we all want tablets that convert to laptops; laptops that convert to desktops; smartphones that convert to pizza ovens; or whatever. The latest in the convertible category from Toshiba (www.toshiba.com) is the Satellite Click – a Windows 8 tablet/laptop said to be the first detachable device to utilize AMD's low power A4 processor, plus AMD Radeon 8000 graphics and 4 GB of 1,600 MHz DDR3L RAM. It does come with some standard tablet amenities, including a built-in webcam, Micro USB 2.0 and HDMI® ports, a microSD™ slot, plus accelerometer and gyroscope sensors. When you combine the 13.3 inch multitouch screen with the full-size keyboard and touch pad, you're back in laptop land.

The Click uses a dual battery setup, so you can charge either piece while docked or undocked. A 500 GB hard drive is built into the tablet section. As of this writing, the Click is available only through Best Buy stores or ToshibaDirect.com with a base price of \$599.99. ▲



■ Toshiba's Satellite Click, a tablet and laptop in one.

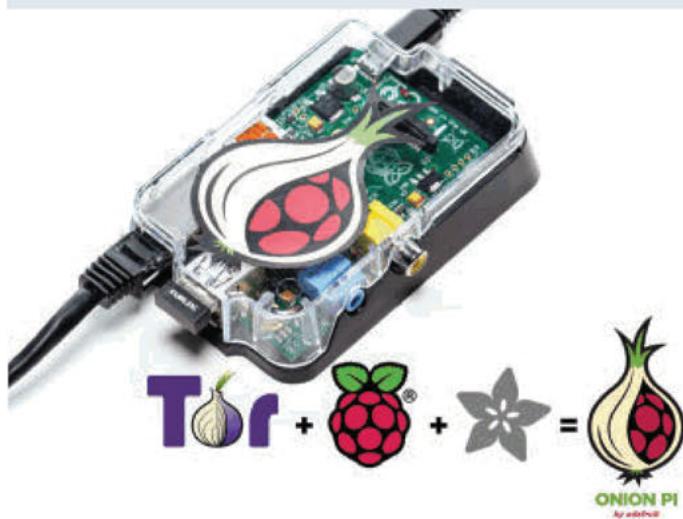
Build a Secure Wireless Access

If you feel the need to prevent covert operators from snooping into your online activities, your best bet is still the Tor anonymity network – which even the NSA and the British equivalent GCHQ have been largely unable to defeat. Just last year, the NSA admitted that Tor is still "the king of high secure, low latency Internet anonymity" and that there "are no contenders for the throne in waiting."

The simplest way is to log onto the Tor Project website (www.torproject.org) and download the browser. So, what if you want to surf anonymously using a netbook, tablet, smartphone, or other device that can't run Tor or doesn't have an Ethernet connection? Or, what if you want to use a computer at a friend's house or at work, where it is not appropriate to install new software? In that case, you might be interested in a Raspberry Pi device offered by the folks at Adafruit Industries (www.adafruit.com).

The Onion Pi Pack offers everything you need to build a Raspberry Pi-based Tor proxy that you can use virtually anywhere, on almost any device. The pack includes a Raspberry Pi Model B, a mini Wi-Fi adapter, a 4 GB SD card with Raspbian 7 already installed, a case, a 5V power adapter, and all the cables you need to create your own portable low power wireless access point.

According to Adafruit, the project is "best used by people with a little bit of command-line, Linux, or Raspberry Pi experience," but they do have "tons" of tutorials available if you get stuck. The kit will set you back \$89.95, but it's a small price to pay for keeping those questionable activities to yourself. ▲



■ Adafruit's Onion Pi pack with mini Wi-Fi.

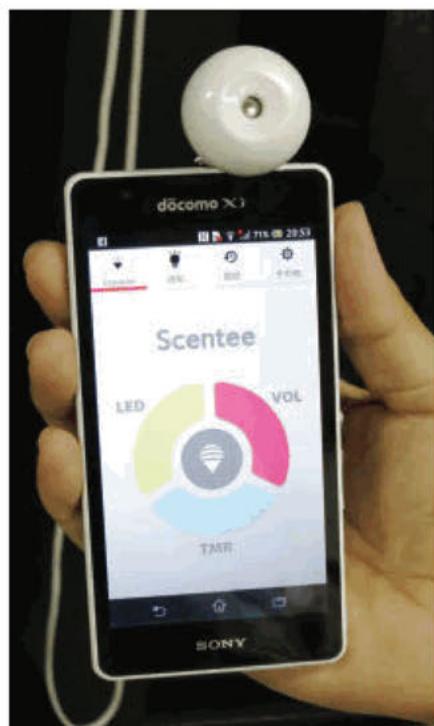
COMPUTERS and NETWORKING Continued

Answer the Smellophone

It was way back in 1960 that Hans Laube created the ill-fated Smell-O-Vision system, designed to inject up to 30 different odors into movie theaters when triggered by the soundtrack of a film. Its one and only implementation was with the film "Scent of Mystery," and it was anything but a roaring success. Like a bad penny, the concept has re-emerged in the form of an iPhone and Android smartphone accessory from Scentee, Inc. (www.scentee.com). The device — also called Scentee — works with an app that tells it when to spray a fragrance into the air, and it can be set to go off when you receive an email, a text message, or a "like" on Facebook — or even at a particular time of day.

Scentee just plugs into the headphone jack and provides up to 100 sprays per \$5 cartridge. The scent options include predictable air freshener aromas such as rose, jasmine, cinnamon roll, strawberry, coconut, and so forth. You can also pick corn soup, Korean BBQ meat, or baked potato, however. Reportedly, a bacon scent is coming soon. Alas, as of this writing, it is available only on Amazon Japan for the equivalent of \$37 including your first scent cartridge. But who knows? It may turn up in the US eventually. ▲

■ The Scentee smartphone accessory tickles your olfactory organs.



Mobile Photography Tools

More useful smartphone add-ons are offered in the form of mobile photography tools from olloclip (www.olliclip.com) — now available for the iPhone 5/5s/4/4s. One option is the Telephoto + Circular Polarizing Lens which could come in handy for trips to the zoo and so on. The \$99.99 price tag gets you only 2x magnification, which doesn't exactly pull in distant objects. For comparison, a \$9.97 pair of Walmart binoculars offers 10x. A better bet might be the new

4-In-One lens which combines a fisheye lens, a wide-angle lens, a 10x macro, and a 15x macro — all for \$69.99. In any case, you just clip the device over the smartphone's camera hole and you're ready to shoot. Visit www.olliclip.com for details. ▲



■ A 10x close-up of a watch face shot using an olloclip macro lens.



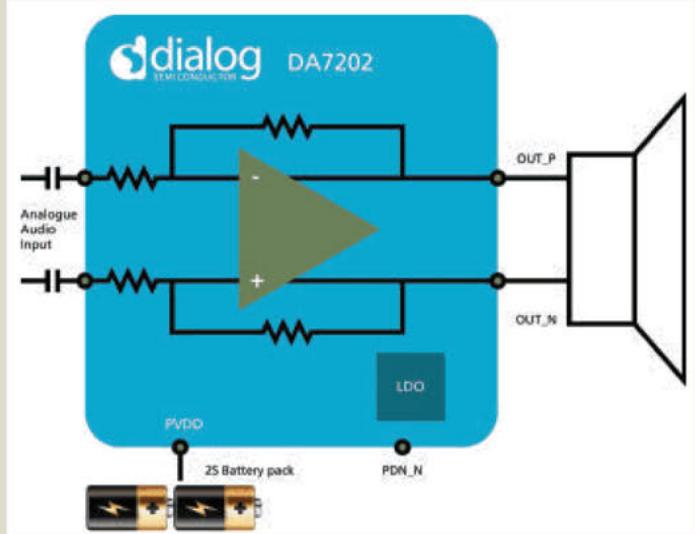
CIRCUITS and DEVICES

Better Ultrabook Audio

One might think that "Ultrabook" is just another meaningless marketing term but, in fact, Intel (which calls the shots) spells out quite a few required specs, including the use of a low voltage Intel Core processor. Unfortunately for music fans, low voltage tends to translate into crummy audio quality. Next-generation Ultrabooks that employ a new speaker driver from Dialog Semiconductor (www.dialog-semiconductor.com) should prove to be more satisfactory, fortunately.

The DA7202 is the first product in the company's range of Class D audio products designed to deliver three times the power output for Ultrabooks, tablets, and various speaker accessories powered by a dual-cell battery pack. Typically, most mobile devices use a CODEC with an integrated Class D amp that can't drive the speakers with more than 1A. The DA7202 will drive up to 3.5 Wrms. In addition, having a direct connection from the battery to the amp eliminates DC-DC power loss.

According to Dialog, the increased power does not compromise sound quality, and the chip provides a 100 dB signal-to-noise ratio. ▲



■ Block diagram of the DA7202.

INDUSTRY and the PROFESSION

Panasonic Bails on Plasma TVs

Those of us who swear by our plasma screen HDTVs will soon be swearing in general, as Panasonic will be shutting down its plasma display panel (PDP) plants by the end of March. In 2010, plasma TVs accounted for about 40 percent of the market, and Panasonic had a 40 percent share of it. However, plasma's share is expected to drop to five percent this year. Reportedly, Panasonic has lost \$15 billion in PDP display production and will refocus on products outside the TV market. NV

NEW! from Velleman®

3D PRINTER KIT
Model K8200

Advanced kit requires mechanical, electronics and computer skills

Build-it-yourself 3D printer. Prints objects of maximum size of 20 x 20 x 20 cm using either PLA or ABS filament (3mm.) Extremely fast, reliable and precise even when printing at high speeds. Model K8200 is compatible with RepRap software and firmware (Free download). Sturdy aluminum construction, with heated print bed. Nine different colors of PLA filament available at \$39.00. Complete with power supply and all necessary parts. Filament sold separately.

NOW ONLY \$795.00
Part No. 32VKK8200M

Electronix EXPRESS

TERMS: Min. \$20 + shipping. School Purchase Orders, VISA/ MC, Money Order, Prepaid. NO PERSONAL CHECKS, NO COD. NJ Residents: Add 7% Sales Tax.

In NJ: 732-381-8020
FAX: 732-381-1006

900 Hart Street • Rahway, NJ 07065
800-972-2225

<http://www.elexp.com>
email: electron@elexp.com

Shifty Business

With the ever-increasing capabilities and sophistication of modern microcontrollers, it's very easy to overlook tried-and-true solutions from the past. We shouldn't, though, lest we miss an opportunity to have "small" technology make a big impact on a project. Case in point: shift registers. No matter how many I/O pins our favorite processor sports, we always seem to want or need more. This is exactly where shift registers can save the day — with simple connections and equally simple software to use them.

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/january2014_SpinZone.

I tend to design small discrete projects, so when the BASIC Stamp 1 was all I had, I was happy with eight I/O pins. Then came the BS2 and twice as many. The SX28 came with four more, and the SX48 with more still. With the Propeller, we have 28 I/O pins that are completely free for our use (two pins are used for the EEPROM, and two pins for the programming/debug connection).

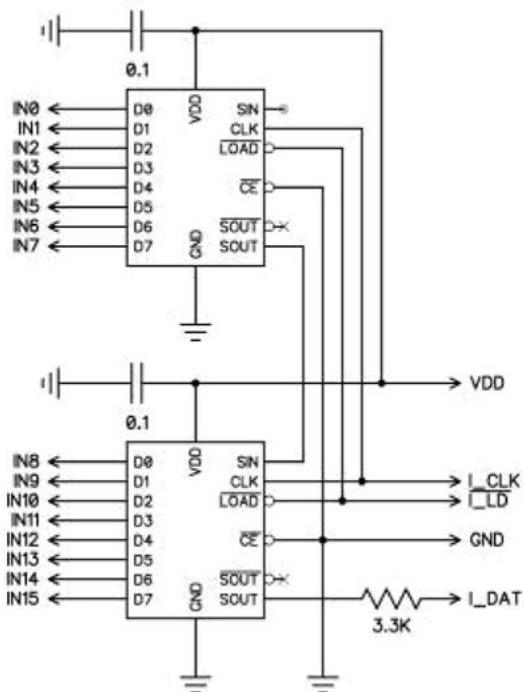
While 28 pins is usually enough for what I do, I often assist friends with greater I/O ambitions than my own. A few years ago, my friend Matt Hawkins (www.likeform.com) — an amazing animatronics creator — asked me to help him put together a Propeller-based I/O board that he could use in conjunction with large-scale professional show-control systems. He needed RS-232 (for coms with the show controller), RS-485 (for DMX output), 16 inputs, 16 outputs, a PS/2 port for a trackball, and a microSD adapter to store output patterns that could be called from code.

We've obviously blown way past the 28 pins available on a Propeller. The solution to the requirement of 16 inputs and 16 outputs was to use shift registers; what would have been 32 discrete pins is boiled down to six with a little code help. With two 74HC165 shift registers and just three I/O pins, the board can read 16 digital inputs. Likewise, with two 74HC595 shift registers and three I/O pins, the board can control 16 digital outputs. I keep the input and output pin groups separated so their respective code can run in different cogs if needed.

Shifty Inputs

We tend to think of computers as INPUT-PROCESSING-OUTPUT, so let's have a look at expanding digital inputs using the 74HC165 shift register. The '165 has eight inputs. There is a pin called Shift/Load that is used to capture the state of the inputs to an internal holding register. When we do this, the MSB (most Significant Bit) of the inputs is placed on the Q_h pin (labeled SOUT in my diagrams). We can read this with a Propeller I/O pin. By taking the clock line high then back low, the next bit is output to the Q_h pin. By doing this eight times, we can read all the inputs.

Figure 1 illustrates the connections for using two 74HC165s as we did on Matt's board. Note that the



■ FIGURE 1. 74HC165 inputs.

clock and shift/load lines are tied together. The interesting bit is that the serial output from the upper chip feeds a serial input pin on the lower chip. These connections essentially create one 16-bit shift register from two '165s. Of course, we can extend this to 24 or even 32 inputs if desired. It is electrically possible to have more than four '165s connected in a chain but I never do that — it doesn't make sense given the 32-bit values used by the Propeller.

If you look closely, you can see that input 15 is nearest to the Qh (SOUT) pin used as the I_DAT connection to the Propeller; if we add another '165, that will change to input 23 (MSB of a 24-bit group), etc. Keep this in mind when planning your designs and input assignments; the inputs are shifted into the Propeller MSB first.

Finally, the purpose of the 3.3K resistor in the SOUT line is to allow a VDD of five volts. The 3.3K resistor limits the current into the 3.3V Propeller pin to a safe level.

For those with PBASIC or SX/B experience, you're familiar with the **SHIFTIN** command. We don't have that in the Spin language, so we'll use a bit of code to create its equivalent. The object code for reading from one to four 74HC165s consists of a whopping two methods. That's right ... two. Some newcomers might think why bother putting so little code into a separate object file. The point of an object file is to simplify code re-use; copy-and-paste is for the birds!

The **start()** method takes our I/O pin assignments and configures them for use:

```
pub start(dpин, cpин, lpин)
    longmove(@dta, @dpин, 3)

    dira[dta] := 0
    outa[clk] := 0
    dira[clk] := 1

    outa[load] := 1
    dira[load] := 1
```

This method takes three parameters: the data pin, the clock pin, and the shift/load pin. The pins are copied to global variables within the object using **longmove**. There is no error checking for valid pin numbers, so do be mindful of your assignments.

The data pin is configured as an input, the clock pin as an output/low, and the shift/load line as an output/high. That last section puts the '165 chain into shift mode where data can be transferred from the holding registers into the Propeller.

Now for the action. The **read()** method will allow us

to read from a chain of one to four 74HC165s. The method takes two parameters: the bit orientation (MSBFIRST or LSBFIRST) and the number of devices connected:

```
pub read(mode, count) | in165
    count <= 3

    outa[load] := 0
    outa[load] := 1

    in165 := 0
    repeat count
        in165 := (in165 << 1) | ina[dta]
        outa[clk] := 1
        outa[clk] := 0

    if (mode == LSBFIRST)
        in165 >= count

    return in165
```

At the top of the method, we convert the byte count to bits by multiplying by eight. As eight is a power of two, we use a left shift instead of multiplication. Shifting is much faster when we can use it.

The shift/load line is taken low to transfer the inputs to the internal holding register, then returned high to put the '165 back into shift mode.

The inputs will be collected in a variable called *in165*. This is a local variable, so we need to clear it to zero before use. Local variables come from the stack and could have anything in them (locals are not initialized to zero like globals).

A **repeat** loop takes care of grabbing the bits. Note that the first line of the loop is doing two things: It starts by left-shifting the value in *in165* to make room for the new bit; then it moves the state of the data pin to the *in165.bit0*. The clock line is taken high, then back low to get the next bit.

The normal configuration for using 74HC165s is MSBFIRST (1) as in **Figure 1**. Let's say for the sake of a circuit you wanted to flip those on end, such that what was input 15 becomes input 0, and vice-versa. No problem! Simply set the mode to LSBFIRST (0). The bits will be flipped by using Spin's reverse operator ($>\<$). As you can see, the mode affects all bits returned.

What if we just wanted to flip the lower eight bits of the inputs? We can do that; it just takes one more line of code:

```
buttons := ins.read(ins#MSBFIRST, 2)
buttons.byte[0] >= 8
```

The first line reads two bytes (16 bits) into `buttons` using MSBFIRST mode. The next line reverses the bits in byte0 of `buttons`; bit0 and bit7 are swapped; bit1 and bit6 are swapped, etc. Be a little careful with the reverse operator; it clears unused bits to zero. In the above code, we affected all the bits in `buttons.byte0`, so we're fine. If we wanted to flip the lower four bits of `buttons`, we would have to do it like this:

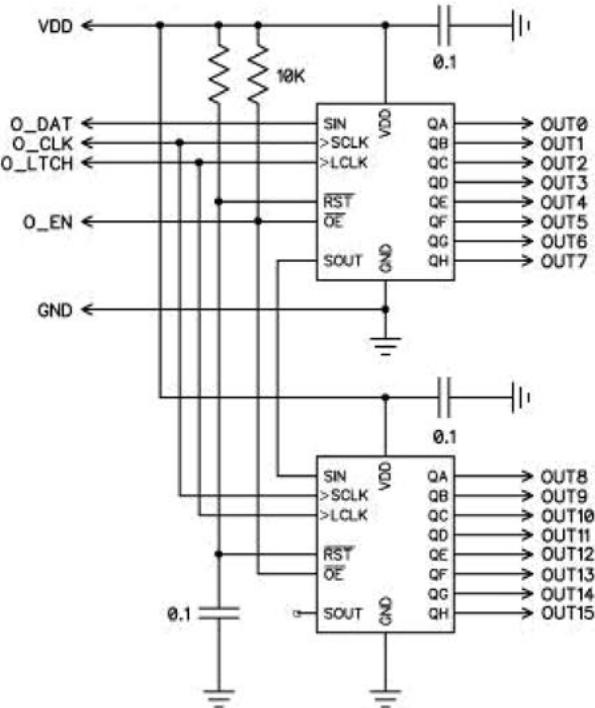
```
buttons := ins.read(ins#MSBFIRST, 2)
buttons := (buttons & $FFF0) |
(buttons.byte[0] >< 4)
```

Shifty Outputs

There is a nifty companion to the 74HC165 called the 74HC595 that allows us to convert three (sometimes four) I/O pins into eight outputs. Or 16. Or 24. Or even 32! **Figure 2** illustrates the use of two 74HC595 chips to create 16 digital outputs.

Let me point out that this is my "industrial" version. Here, the Output Enable pins are pulled up to disable (tri-state) the outputs from the '595s. This configuration gives the Propeller plenty of time to boot up and preset the outputs. By pulling the OE line low, the outputs will go active. When using this configuration, you will want to pull your '595 outputs to a known state.

At EFX-TEK, we were recently contracted to build a



■ FIGURE 2. 74HC595 outputs.

controller for road signs. We used the '595 circuit as shown to drive the gates of MOSFETs for the controller outputs. The gate of each MOSFET was pulled low to ensure the output would stay off until we wanted it to be on ('595 outputs enable and high).

If you don't need to control the Output Enable, remove the pull-up from that line and tie the OE pin(s) to ground.

Operation of the '595 is a mirror of the '165. The Propeller will place a bit on the serial input pin, then take the clock high and back low to move that bit into the '595 holding register. When all bits are shifted in, the Latch line is pulsed high, then back low to transfer the holding register to the outputs. The advantage of the holding register is that outputs do not ripple as we're shifting the data.

In **Figure 2**, you can see that the serial output of the first device is connected to the serial input of the second. It's important to understand that the bits for the second device are forced through the first device. When cascading chips like this, we need to know how many devices are in the chain to set the outputs correctly (of course, this applies to the '165 as well).

Like the object for the '165, the code for the '595 is super easy. There are, in fact, two more methods in this one, but those are used to set and clear the Output Enable line if it is used.

As before, the `start()` method takes the pins and configures them for use:

```
pub start(dpin, cpin, lpin, epin)

longmove(@dout, @dpin, 4)

dira[dout] := 1

outa[clk] := 0
dira[clk] := 1

outa[latch] := 0
dira[latch] := 1

if (eout => 0)
    outa[eout] := 1
    dira[eout] := 1
```

After copying the pin numbers to global variables, they are configured as outputs. The clock and latch line are set up as output and low. If the Output Enable pin is used, it is set up as an output and high to disable the outputs from the '595s. If we don't have an Output Enable line in the circuit, we pass -1 as the `epin` parameter.

The `write()` method will move from one to four bytes to a 74HC595 chain:

```

pub write(out595, mode, count)

count <= 3

if (mode == LSBFIRST)
    out595 >= count

out595 <= (32 - count)

repeat count
    outa[dout] := (out595 <= 1)
    outa[clk] := 1
    outa[clk] := 0

    outa[latch] := 1
    outa[latch] := 0

```

As before, the byte count is converted to bits. If LSBFIRST mode is specified, the bits to be output are reversed. In preparation for output, the bits are shifted left such that the MSB bit of the desired output lands in *out595.bit31*. We do this because the behavior of the shift loop is LSBFIRST.

The first line of the loop uses the rotate left (<) operator to move bit31 into bit0. This is what lands on the data pin which has been set to output mode. By taking the clock line high and then back low, the bit is moved into the '595. When all bits have been shifted, the latch line is taken high and then back low to transfer the '595 holding registers to the outputs.

Easy peasy!

Before moving on, let me point out that if you want to run either of these objects in a secondary Spin cog, you can, but you must call the *start()* method from inside that cog. This is necessary to properly set up the pins (via the *dira* register) for the cog using the code.

A New Trick for an Old Dog

After we got Matt's board working (**Figure 3**), he called me one day and asked, "Can we do PWM on the outputs?" My typical response normally would have been, "I don't know ..." but, in fact, I had seen a PWM object for '595s in the Object Exchange. To be honest,

it didn't make a lot of sense to me but since someone had taken the time to post it, I assumed it worked – then, of course, I set out to write my own!

I've written lots of PWM code for the Propeller and the *pseudo-code* for it looks like this:

```

repeat cycle from 0 to 255
    if (level > 0) and (cycle <= level)
        output is on
    else
        output is off

```

The trick here is to do this for 16 levels, write the output bits into a holding variable, then shift that holding variable out to the '595s. It turns out that with the power and flexibility of PASM, the code is pretty simple.

The *start()* method for the PWM object is identical to the normal PWM object – the difference being that the pins will be converted to masks for use with PASM, and then that code will be launched into its own cog:

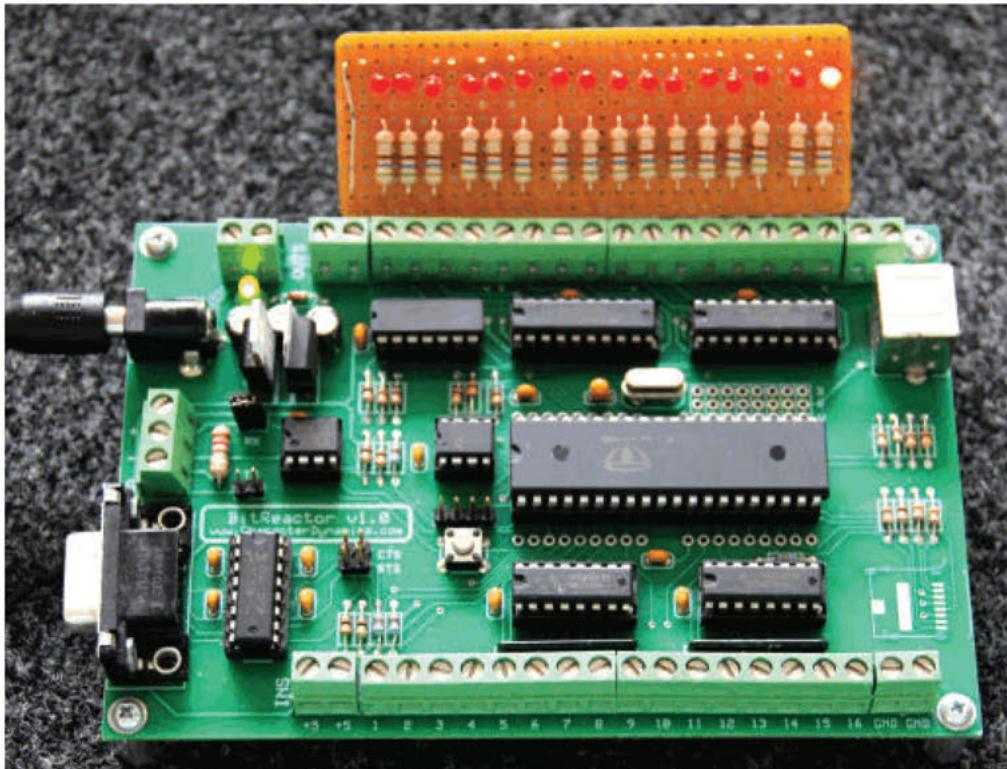
```

pub start(dpin, cpin, lpin, epin)

stop

odatmask := 1 << dpin
oclkmask := 1 << cpin
olatchmask := 1 << lpin

```



■ FIGURE 3. Smart ins and outs.

```

eout := epin
if (eout => 0)
    outa[eout] := 1
    dira[eout] := 1

hubaddr := @brightness

cog := cognew(@entry, 0) + 1

return cog

```

Long-time readers will notice that I've reverted to updating the PASM code before launching. The reason is that it's cleaner for Spin programs, and I can't easily use the PASM code with other languages. That is to say that I've stopped jumping through hoops trying to make my PASM sections portable. Speaking of PASM, let's jump in now that it's launched:

```

dat
    org 0
entry
    mov outa, #0

```

```

        or      dira, odatmask
        or      dira, oclkmask
        or      dira, olatchmask

```

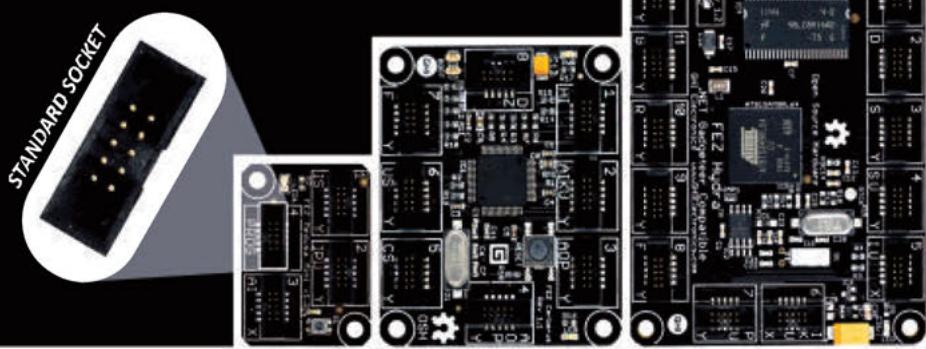
At the top, we clear the pins and then use the masks to set the data, clock, and latch pins to the output state. Now, we can drop into the PWM loop:

pwmmain	mov hub, hubaddr
	mov chcount, #16
	mov outbits, #0
	mov chmask, #1
:loop	rdbYTE t1, hub wz
if_z	jmp #:next
if_be	cmp cycle, t1 wc, wz
	or outbits, chmask
:next	add hub, #1
	shl chmask, #1
	djnz chcount, #:loop

We start this section by making a copy of the hub address of the brightness array. The [fixed] channel count

SOLDERLESS ENGINEERING from **BEGINNER** to **ADVANCED**

Choose from a
VARIETY OF BOARDS



Select an IDE

Arduino IDE
C and C++

Visual Studio
C# and VB

Connect **100+ options** to create all kinds of gadgets.

Get 10% OFF with code **NVGET10** at
www.GHIElectronics.com

BILL OF MATERIALS

74HC165	Mouser 595-SN74HC165N
74HC595	Mouser 595-SN74HC595N
TPIC6A595	Mouser 595-TPIC6A595NE
0.1 µF	Mouser 80-C315C104M5U
3.3K	Mouser 299-3.3K-RC
10K	Mouser 299-10K-RC

is set to 16; the working outputs (*outbits*) are cleared to zero; and, finally, the channel bit mask is set to one for *outbits.bit0*.

At :loop, we read a byte from the brightness array and allow the rdbyte instruction to affect the Z flag. If the level for that channel is zero, the Z flag will be set. The next line looks at that flag and if the channel level is zero (Z flag = 1), we jump to the section at :next.

If not zero, we compare the cycle position (0 to 255) with the channel level. If the cycle position is below or equal (if_be) to the channel brightness, we set the channel bit to turn on the output (we don't have to clear the channel for off as *outbits* starts at zero at the top of

the PWM loop). At :next, we advance the hub pointer and shift the channel bit mask to the left. When all channels have been analyzed, we drop through to the shifty section:

```
shiftout16    mov   bitcount, #16
               shl   outbits, #(32-16)

:loop         shl   outbits, #1 wc
               muxc outa, odatmask
               nop
               or    outa, oclkmask
               nop
               andn outa, oclkmask
               djnz bitcount, #:loop
               or    outa, olatchmask
               nop
               andn outa, olatchmask
               add   cycle, #1
               and   cycle, #$FF
               jmp   #pwmmain
```



We Provide Complete Solutions
for Embedded Systems

www.embeddedARM.com

(480) 837-5200

- 30 years in business
- Never discontinued a product
- Open Source Vision
- Engineers on Tech Support
- 45 Day Evaluation Period

Single Board Computers

TS-7250-V2



pricing
starts at
\$199
qty 1
\$165
qty 100

- Up to 1GHz ARM9
- 512MB RAM
- 4GB eMMC Flash
- Ethernet, USB, CAN, DIO, PC/104
- Low power and sleep mode
- Runs Debian Linux

Industrial Controllers

Rugged, Affordable, Flexible



Controllers
start at
\$199
qty 1
Modbus
Peripherals
start at
\$119
qty 100

- Up to 1GHz CPU
- Linux Fast Boot
- Opto-isolated ports
- Screw-down connectors
- Counters & quad decoder
- Debian and GNU tools

Modbus Peripherals

- TS-1400 6 relays
- TS-1420 12 outputs
- TS-1700 DIO, Temp Sensor
- TS-1800 14 A/D, 24 DIO

With the exception of being fixed at 16 bits, this code is just like the Spin code to transfer bits to the '595s.

For MSBFIRST output, we shift *outbits* left by 16 to position our MSB in *outbits.bit31*. At :loop, we shift *outbits* to the left by one, capturing bit31 in the Carry flag. We transfer the Carry flag to the data output using **muxc** and the data pin mask. A **nop** is inserted to let the data pin settle before bringing the clock pin high. Again,

a **nop** is used to give the clock pulse a bit of weight before pulling it back low.

This process is repeated for all bits. When all have been shifted, the latch pin is taken high, then back low — you've seen this before.

The cycle counter is incremented and ANDed with \$FF to keep 0 to 255. At this point, we jump back to the top and do it all over again.

In this object, we're not using discrete bits, but DMX-compatible byte values (0 to 255) for each output, so we have to run the PWM loop 256 times. There is a little bit of loop "wobble" due to the way a zero level is handled, but for LEDs I don't see this as a problem. A quick test showed that the PWM frequency is better than 150 Hz.

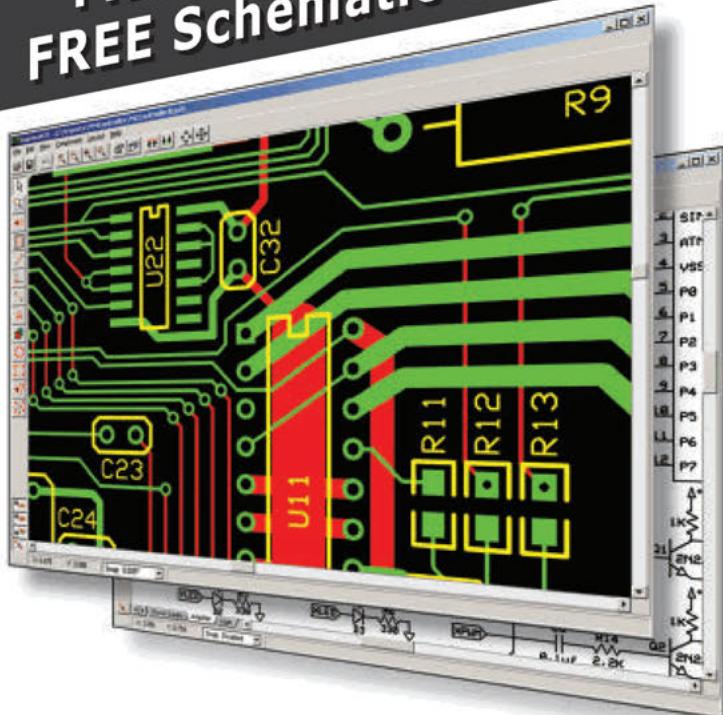
More Oomph for Your Outputs

I fibbed a bit about Matt's board. We're not actually using 74HC595 shift registers on it. What we are using is the signal-compatible TPIC6A595. From a code standpoint, this chip works just like the 74HC595 — even with PWM. The difference is in the outputs. The TPIC6A595 uses open-drain MOSFET outputs that are capable of handling up to 50V at 350 mA each, and they cannot be tri-stated. This chip is well-suited for controlling relays and solenoids, hence found in a lot of industrial applications.

Okay ... as if you needed an excuse to hook-up and control that box full of LEDs you've got sitting on your workbench — now you have a way to do it without using up all your Propeller I/O pins.

Happy New Year! Until next time, keep spinning and winning with the Propeller! **NV**

\$51 For³ PCBs
FREE Layout Software!
FREE Schematic Software!



- 01 DOWNLOAD our free CAD software
- 02 DESIGN your two or four layer PC board
- 03 SEND us your design with just a click
- 04 RECEIVE top quality boards in just days

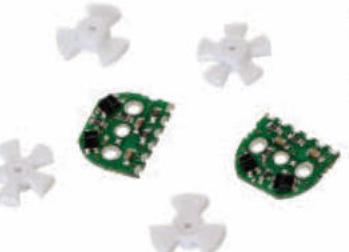
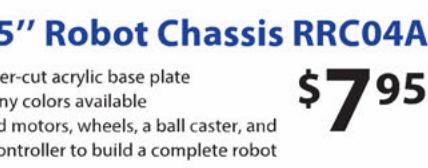
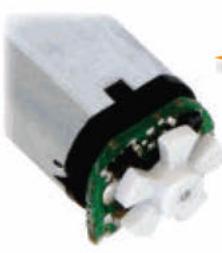
expresspcb.com

RESOURCES

Jon "JonnyMac" McPhalen
jon@jonmcpalen.com

Parallax, Inc.
www.parallax.com
Propeller chips and
programming tools

Finding the right parts for your design can be difficult, but you also don't want to spend all your time reinventing the wheel (or motor controller). That's where we come in: Pololu has the unique products — from actuators to wireless modules — that can help you take your design from idea to reality.

 <p>170-Point Breadboard ITEM #1486 \$2.95</p> <p><i>Other colors available</i></p> <ul style="list-style-type: none"> • 17 rows of tie points for up to two 14- or 16-pin DIP ICs • Tabs allow multiple units to be connected for larger projects 	 <p>Wall Power Adapters STARTING AT \$5.95</p> <ul style="list-style-type: none"> • Center positive, 5.5 mm × 2.1 mm barrel connector • Available voltages: 5 V, 9 V, and 12 V • Available currents: 1 A, 3 A, and 5 A 	 <p>ITEM #2135 \$4.49</p> <ul style="list-style-type: none"> • DRV8835 driver • 2 V to 11 V operating range • 1.2 A continuous (1.5 A peak) per channel • 14-pin DIP form factor 	 <p>ITEM #713 \$4.95</p> <ul style="list-style-type: none"> • TB6612FNG driver • 4.5 V to 13.5 V operating range • 1 A continuous (3 A peak) per channel 	 <p>Dual Motor Drivers ITEM #2130 \$4.95</p> <ul style="list-style-type: none"> • DRV8833 driver • 2.7 V to 10.8 V operating range • 1.2 A continuous (2 A peak) per channel
 <p>Pololu RC Switch with Relay ITEM #2804 \$9.95</p> <p>Enables easy control of large, electrically isolated loads in radio control systems.</p>	 <p>Linear Actuators STARTING AT \$66.95</p> <p><i>Various sizes available</i></p> <ul style="list-style-type: none"> • 2" to 12" stroke lengths • Multiple speeds available • Potentiometer feedback option 	 <p>QTR-L-1RC Reflectance Sensor ITEM #2455 \$4.95 / 2-pack</p> <p>Add line sensing or proximity/edge detection to your robot with these tiny sensors, capable of measuring infrared reflectance at distances of up to 1 inch.</p>	 <p>Pololu 4-Channel RC Servo Multiplexer ITEM #2806 \$9.95</p> <p>Allows for easy switching between two independent RC signal sources.</p>	
 <p>Jrk 21v3 USB Motor Controller ITEM #1392 \$49.95</p> <p>Highly configurable motor controller that offers four control interfaces and can optionally be used with feedback for closed-loop speed or position control. Great for use with our linear actuators or making your own servo!</p>	 <p>LSM303D 3D Compass and Accelerometer Carrier with Voltage Regulator ITEM #2127 \$14.95</p> <p>Combines a digital 3-axis accelerometer and 3-axis magnetometer into a single package that is ideal for making a tilt-compensated compass.</p>	 <p>Optical Encoder Pair Kit for Micro Metal Gearmotors ITEM #2590 \$8.95</p>		
 <p>Dagu Wild Thumper 6WD Chassis (With 75:1 Steel Gearboxes) ITEM #1563 \$249.95</p> <ul style="list-style-type: none"> • Rugged aluminum body • All-terrain suspension • Available in 4WD and other gear ratios and colors 	<p>Compatible Gearmotors \$16.95</p> 			
<p>Dagu Rover 5 Tracked Chassis ITEM #1550 \$49.95</p> <ul style="list-style-type: none"> • Durable rubber tracks • Adjustable ground clearance • Available with encoders 	 <p>5" Robot Chassis RRC04A ITEM #1551 \$7.95</p> <ul style="list-style-type: none"> • Laser-cut acrylic base plate • Many colors available • Add motors, wheels, a ball caster, and a controller to build a complete robot 	 <p><i>Extends only 5 mm past motor</i></p>	<p>Add quadrature encoding to your micro metal gearmotors!</p> <ul style="list-style-type: none"> • 3- and 5-tooth wheels included • 3.3 V and 5 V versions available • Works with our growing assortment of gearmotors with extended back shaft 	

In this column, Russ answers questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. Send all questions and comments to: **Q&A@nutsvolts.com**

- **LED Driver**
- **Remote Control**
- **Video and Power Over Cable**

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/january2014_QA.

LED Driver

QI want to drive a 12 VDC one amp 10 watt LED. I am using a 2N3903 transistor for the output. Can you design a circuit for me using a MOSFET or bipolar transistor? I would like to try a solid-state device instead of a relay. I thought about using a PN2222A transistor, but I don't think it has enough power to handle the LED.

— Jeff Miller

AYou are right about the PN2222A, it is rated at 600 mA; normally, you would want a transistor rated several times the load current of one amp. The device that I chose is rated at six amps and is in a tiny six lead package. Four of the leads are connected to the drain to maximize heat dissipation. The schematic is shown in **Figure 1**; all the parts are surface-mount and are hand solderable.

The MOSFET is FDC637AN (512- is a Mouser part number). It is rated 20V, six amps, and the gate is rated 8V max, so I put a 6.2V zener at the gate to limit the voltage. The ON resistance is .024 ohms, so the power dissipation at one amp will be 24 milliwatts.

Probably don't need any heatsinking but I did add

some copper to the layout just in case; see **Figure 2**. The layout is 0.95 x 0.6 inches (0.57 sq in).

Custom VCXO

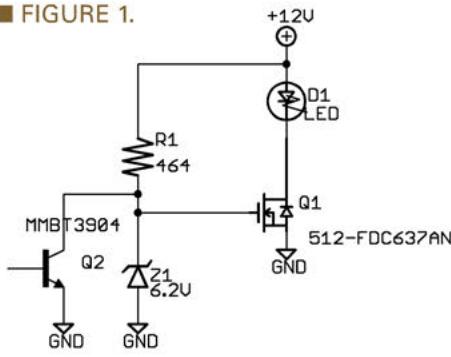
QI need a frequency of 49.848230 kHz for my project. I bought a crystal from Jan Crystals but they sent one for 49.848 MHz third overtone, which oscillates at 16.616 MHz at the fundamental. I think a VCXO will be needed because I want to be exactly on frequency. Can you come up with a design?

— James New

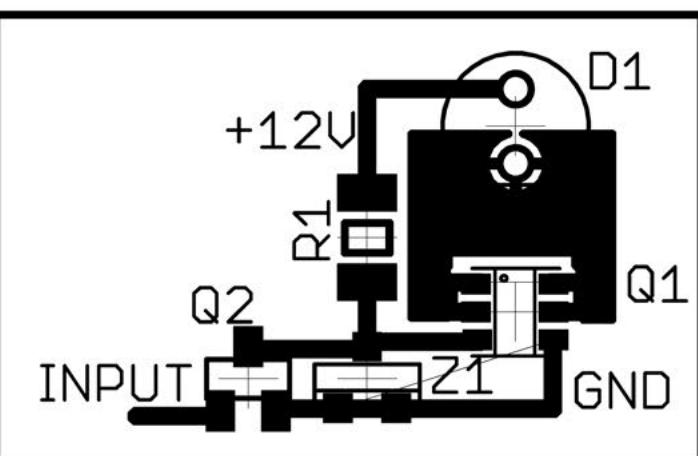
AI didn't want to deal with Jan Crystals to replace the crystal, because I want to divide from a higher frequency — but not that high. The oscillator will have some second harmonic distortion but the divided square wave will have zero second harmonic. I will low-pass filter the square wave with an elliptic filter that has a notch at the third harmonic, so the first visible harmonic will be the fifth.

The filter response at the fifth harmonic (249 kHz) will be far down, so there will be essentially no harmonics. I chose a frequency of 1.5951433 MHz which is 32 times

■ FIGURE 1.

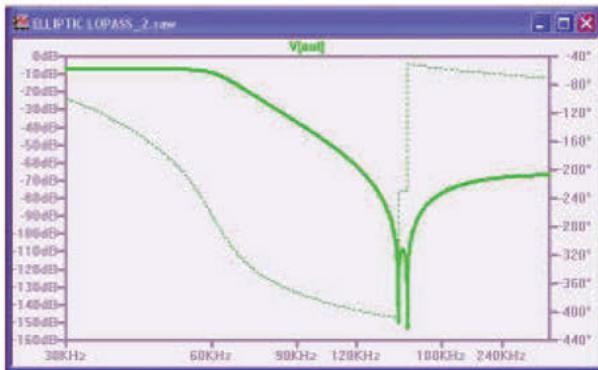


■ FIGURE 2.



QUESTIONS and ANSWERS

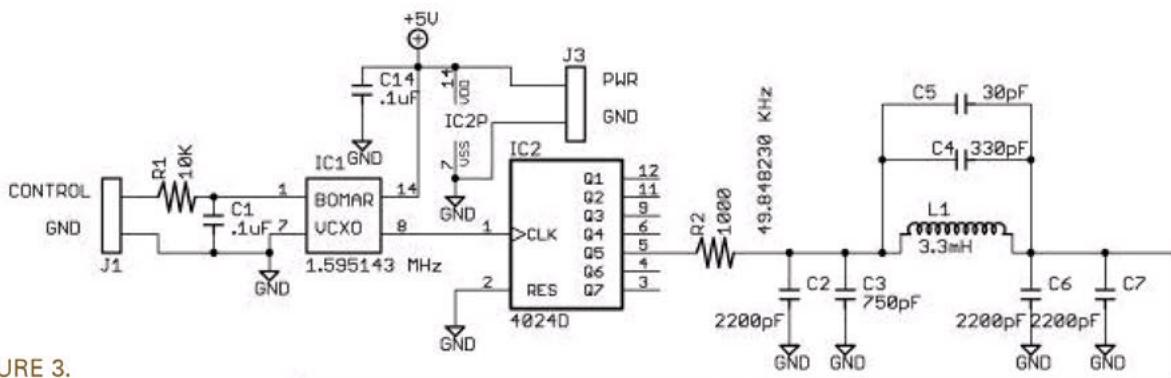
■ FIGURE 5.



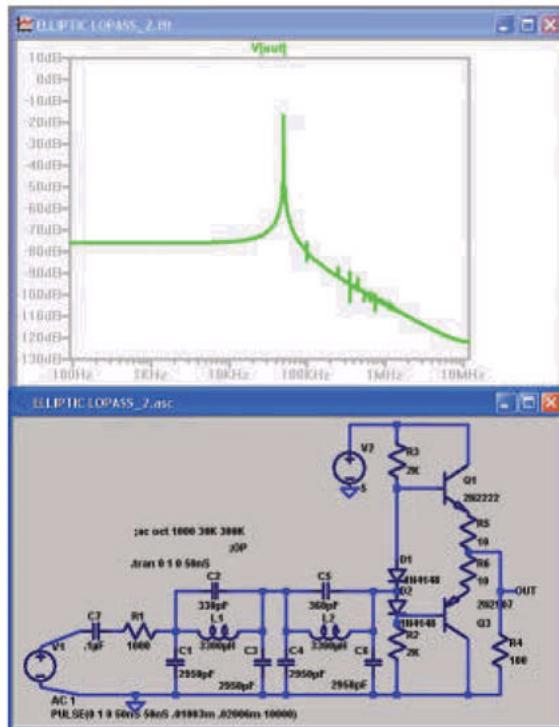
the frequency. I only need to take the fifth output of a seven-stage counter to get 49.84230 kHz.

Bomar Crystal Company has agreed to make the VCXO in a 14-pin DIP package which I incorporated into the design shown in **Figure 3**. I was going to wind my own inductors using ferrite cores that I bought on eBay (got hundreds of them for \$20!), but found that the filter worked fine with 5% inductors which I could buy for 47 cents each. Plus, they were much smaller than ones I could build. The capacitors are low temperature drift (COG) and I provided for trim caps if needed. (See the schematic in **Figure 3**.) The power supply is five volts, and must be stable and low noise to maintain a low noise oscillator signal. The tuning voltage must also be low noise; perhaps a battery would be needed. I added an emitter-follower stage so the filter will not be affected by the load impedance.

■ FIGURE 3.

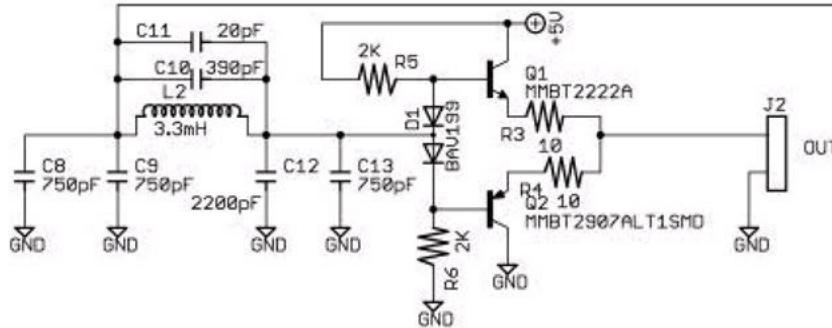


■ FIGURE 4.



I analyzed the filter using LTSpice; see **Figure 4**. The Fourier transform shows some harmonics, but there is a 100 ohms load on it; it should be better with a lighter load.

Figure 5 is the typical response.



Remote Control

QI'm adding a remote control to my power boat winch. I decided to use a 12 VDC remote control receiver. My question: Is it a good idea to use a DC-to-DC 12 volt to 12 volt DC isolated converter to power the remote? Or, can I wire it directly to the battery? Is there an advantage to use the converter?

Thanks in advance.

— Jeff Miller

AThanks for the question. You should have no problems with the DC-to-DC converter. I would not advise directly connecting it to the battery, unless the remote is designed to tolerate the transients and voltage changes caused by the winch. An L-C filter may be sufficient if the remote will operate from 10 volts to 14 volts.

If I knew the battery size and rating and the winch load, I could give a more detailed answer.

The Convenient All-in-One Solution for Custom-Designed Front Panels & Enclosures



You design it
to your specifications using
our FREE CAD software,
Front Panel Designer



ONLY \$90.24
with custom
logo engraving

We machine it
and ship to you a
professionally finished product,
no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

**FRONT PANEL
EXPRESS**

FrontPanelExpress.com
1(800)FPE-9060

Video and Power Over Cable

QMaybe you could be of assistance with my design problem. My objective is to send power along with a composite monochrome (B&W) video signal over a 50 meter (164') length of RG-6 coaxial cable to a 12" LCD monitor that requires a 12 VDC at one amp (12 watt) power source. I would prefer to have the circuit provide at least 1.5 amps at 12 VDC for headroom.

I plan to use a 36 VDC switching power supply as the power source, and a Mean Well model SDM30-24S12 DC-to-DC converter (36 VDC max input) at the monitor end. I have tested this circuit and it works reliably.

My question is how to modulate a composite monochrome video signal onto the 36 VDC source, then demodulate or extract the video and provide a one volt P-P signal into the 75 ohm input of the monitor. Commercially produced systems are available to power remote cameras over coax, but I couldn't find anything to meet my needs. Linear Technology Application Note 87

PCB-POOL
Beta LAYOUT

PCB Prototypes & small series

FREE SMT Stencil
with every prototype order

Embedded RFID
authenticate, track & protect
your product

www.magic-pcb.com

Call Tony: 1 707 447 7744
sales@pcb-pool.com

PCB-POOL® is a registered trademark of

Beta
LAYOUT create electronics

www.pcb-pool.com

"Send Camera Power and Video on the Same Coax Cable" (pages 64 and 65; <http://cds.linear.com/docs/en/application-note/an87f.pdf>) gives a description and schematic of a circuit which may possibly be used as a guide to build a circuit to the specifications I provide above. I would like to keep the circuit as simple as possible with as few components as necessary to provide an image with good resolution.

I have enjoyed reading *Nuts & Volts* for many years; keep up the good work.

— Jeff Hollinshead

A Video cable with included power wire is available. Why not go that route? Check out www.amazon.com/VideoSecu-Security-Camera-Surveillance-Installation/dp/B0017KZ8Y0/ref=sr_1_7?ie=UTF8&qid=1383847027&sr=8-7&keywords=surveillance+system.

Your specs indicate that the monitor is remote and power is supplied at the camera end; that is the opposite of the usual configuration. Why not make the camera remote and use the application note schematic? **NV**

Can't figure out that pesky circuit or don't understand the components?

Let Russ help!

Send any questions and/or comments to:

Q&A@nutsvolts.com

Did you know that if you're a paid subscriber to *Nuts & Volts*, you can get the online version for **FREE**? Get details at www.nutsvolts.com

DIY NEVER HAD A BETTER FRIEND...

Now available at  RadioShack.

Model 301 640-0178

Model 324 640-0177

Take your DIY project to the next level with PanaVise work holding tools. Perfect for hobbies, soldering, electronics, crafting & repair, PanaVise tools have hundreds of uses only limited by your imagination.

PanaVise, "better than a third hand!"

PANAVISE®
Innovative Holding Solutions

www.radioshack.com/StoreLocator

ezPICK VACUUM PICK AND PLACE SYSTEM



ABACOM TECHNOLOGIES

www.abacom-tech.com

NEW PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS

VERY LOW-LOSS MATERIALS FOR AUTOMOTIVE SAFETY AND DRIVER ASSIST SYSTEMS

Isola Group S.p.A. has announced Astra™ – the company's breakthrough very low-loss dielectric constant (Dk) product for millimeter wave frequencies and beyond. Astra revolutionizes RF and microwave designs, as it delivers a thermoset solution which is very easy to process and has stable electrical properties over a wide range of temperatures and frequencies.



The lead-free Astra laminate materials exhibit electrical properties that are constant over a broad frequency and temperature range. Astra features a Dk that is stable between -55°C and +125°C. In addition, Astra offers a lower dissipation factor (Df) of 0.0017, making it a cost-effective alternative to PTFE and other commercial microwave laminate materials.

Astra does not require the use of plasma cleaning – an offline and expensive PCB (printed circuit board) hole-wall preparation process. Astra enables lower drilling costs, as its unfilled system provides easier drilling and extends drill life. Astra demonstrates a high-peel strength which enables use of special copper types to deliver very low passive

intermodulation numbers.

Astra is suitable for RF/microwave printed circuit designs that operate at 24 GHz and 77 GHz frequencies. Key applications include long antennas and such radar applications for automobiles as adaptive cruise control, collision avoidance, blind spot detection, lane departure warning, and stop-and-go systems.

ULTRA-LOW LOSS MATERIALS FOR HIGH SPEED DIGITAL, RF, AND MICROWAVE APPS

Also now available from Isola Group S.p.A. is TerraGreen –

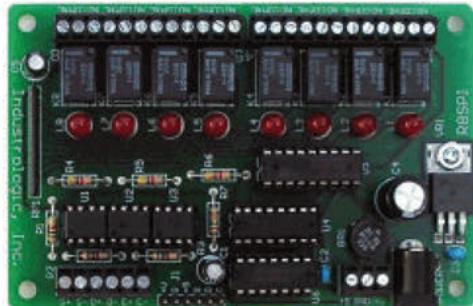
ISOLATED SPI INTERFACE EIGHT RELAY BOARD

The R8SPI now available from Indstrologic is a small printed circuit board (PCB) assembly designed to provide eight medium current relays to microcontroller boards via an opto-isolated SPI style interface. It includes eight one amp SPDT (Form "C") relays with all three relay contacts available, and convenient screw terminal block connections for all relay, SPI, and power connections. The R8SPI can be controlled by opto-isolated inputs for the SPI control signals STROBE, DATA, and CLOCK, and also includes alternate non-isolated logic level input SPI controls. It has an onboard +5 volt power supply for the relays and external circuitry, and a pin-type

the company's new halogen-free, ultra-low loss, RF/microwave/high speed material. TerraGreen is engineered for such high performance applications as power amplifier boards for 4G LTE base stations, Internet infrastructure, and cloud computing.



TerraGreen has a full offering of cores and prepreg utilizing spread-glass fabric. The dielectric constant (ranging from 3.00 to 3.45) and dissipation factor (ranging from



power connector for easy connection to wall block power supplies. The R8SPI is available as a complete circuit board assembly, and can be enclosed in an optional ABS plastic enclosure. The R8SPI package is shipped complete with all items necessary to begin using it immediately, including a wall block power supply and a hardware reference manual.

For more information, contact:
Indstrologic, Inc.
www.indstrologic.com

0.0030 to 0.0035) remain stable over a wide range of frequencies and temperatures. Core thicknesses from 0.002" to 0.018", 0.020", 0.030", and 0.060" are available.

TerraGreen is a lead-free assembly material and is easy to process. This high performance material utilizes a short-lamination cycle; the product is easy to drill, does not require plasma desmear, and the prepreg shelf life is similar to FR-4 materials. TerraGreen is suitable for high layer count, high speed digital backplanes, and is compatible with Isola's FR-4 materials for hybrid designs.

For more information, contact:
Isola Group S.à.r.l.
www.isola-group.com

OFF-THE-SHELF COLOR TOUCH DISPLAY MODULES

Reach Technology, Inc., has launched a new product line to help product designers add user interfaces that look like an iPad® or iPhone® with scrolling, sliding, transparencies, 3D graphics, and animations to their products. These modules:

- Include a high level Integrated Development Environment (IDE) with drag-and-drop visual design tools provided by the Qt Project – an open source collaboration used in thousands of commercial products.
- Come with example code, video tutorials, and Reach custom visual components.
- Provide full connectivity options: RS-232, RS-485, CAN bus, Ethernet, USB host, USB device, I²C, SPI, and USART.
- Are field proven with extreme ESD protection and ultra-low EMC for easy regulatory approval.



The G2C1 4.3" display modules are ultra compact and combine a multi-function processor board and a 4.3 inch WQVGA (480 x 272) TFT color touch LCD. The display modules have a wide variety of communications interfaces, eight pins of GPIO, an audio beeper, and a battery-backed real time clock. Onboard Flash memory is used for program and data storage. A "captive" microSD card socket (not the push-pull type that can become dislodged during shipping) is provided for optional mass storage.

Complete development kits are available for \$449 that include all the hardware, software, and support needed to create a prototype in days as opposed to months. The display module included in the kit is ready for production orders and offers 5-7 years availability at a minimum.

For more information, contact:
Reach Technology, Inc.
www.ReachTech.com

i.MX283 EMBEDDED SOLUTIONS

Technologic Systems announces a family of products based on the Freescale i.MX283 ARM9 CPU running Linux at 454 MHz. This release (including multiple form factors) enables a wide range of embedded systems applications and provides flexibility for customers to design a low power, low cost, and long life embedded solution.

Products include:

- TS-4600 computer on module.
- TS-7600 compact single board computer.
- TS-TPC-8390-4600 touch panel computer.
- TS-4600 computer on module (guaranteed available until 2025).

The TS-4600 is a long-life TS-SOCKET computer module that provides a migration path from the TS-4500, offering a performance increase and additional RAM for more flexibility.

The TS-4600 runs at approximately 1.5W base power and features up to 256 MB of DDR2 RAM, 10/100 Ethernet, high speed USB hosts, and two microSD sockets. Additional features include battery backed RTC, SPI, I²C, ADC, eight UART ports, 107 DIOs, and LCD support.

The TS-4600 is available at \$99 (quantity 100). Single unit purchases are also available.

The TS-7600 embedded computer includes a 44-pin header and a 26-pin header with external interfaces for DIO lines, SPI, I²C, UARTs, or custom FPGA logic. The TS-7600 provides a migration path from the TS-7500/TS-7550 offering a performance increase and additional pins for more flexibility. Features include up to 256 MB RAM, 10/100 Ethernet, high speed USB hosts, two microSD sockets, and more.

The TS-7600 is available now at \$95 (quantity 100) with single unit purchases also available.

The TS-TPC-8390-4600 is an open-frame mountable seven inch touch panel computer powered by



the TS-4600 450 MHz ARM computer module. This low power TPC runs Linux and features up to 256 MB RAM, user-programmable FPGA, two Ethernets, USB, ADC, DIO, audio, and more.

The TS-TPC-8390-4600 is available now at \$399 (quantity 100); a single unit purchase is also available.

For more information, contact:
Technologic Systems
www.embeddedARM.com

MINI HANDHELD LIGHT METER

Anheim Scientific has introduced the first model in its new M-Series of mini handheld environmental meters: the M110 mini light meter.

Features include:

- Light Source: fluorescent, metal halide, high-pressure sodium, and incandescent.
- Lux Range: 400, 4,000, 40,000, and 400,000.
- Foot-candle Range: 40, 400, 4,000, and 40,000.
- List price of \$79.

MINI HANDHELD ANEMOMETER

Anheim Scientific has also released the M130 mini anemometer in its M-Series.

Features of the M130 include:

- Wind Speed: Up to 55 mph (90 km/h).
- Units: Speed in m/s, km/h, mph, knots, or ft/min.
- Response: Light air response 0.4 m/s.
- List price of \$84.

The M-Series meters have the following features in common:



- Large 3-3/4 digit display.
- Small 5.3 x 1.9 x 0.2 inch size weighing only 2.8 oz (80 grams).
- Data hold.
- Maximum/average/minimum hold.
- Zero adjustment.
- Low battery indicator.
- Auto power-off and disable function.
- Light weight (less than 9 oz).
- Powered by two AAA batteries.
- Limited two year warranty.

For more information, contact:
Anheim Scientific
www.anheimscientific.com

COMPACT USB OSCILLOSCOPE

Saelig Company, Inc., announces the availability of the new

PicoScope 2000 series oscilloscopes which are 80% smaller than their predecessors; they are similar in size to a passport but 3/4" thick.

Connected to a PC's USB port for power and communication, these oscilloscopes offer bandwidths up to 200 MHz, making them ideal for field use while offering the performance of benchtop scopes.

They feature a sample rate of up to 1 GSa/s with high speed streaming of data up to 1 MSa/s, enabling data captures of up to 100 million samples in length. The series incorporates a built-in 100 MSa/s or 1 GSa/s waveform generator.

PicoScope 2000 series oscilloscopes can produce standard signals such as sine, square, and triangle waveforms with programmable sweep, and can also act as a 12-bit 20 MSa/s full-function arbitrary waveform generator that can reproduce sampled signals.

The free PicoScope software delivers an uncomplicated high-resolution visual display, and it incorporates a range of advanced signal processing features including a spectrum analyzer, automatic measurements with statistics, channel math, reference waveforms, multiple scope and spectrum views, and serial protocol decoding for I²C, CANbus, SPI, I₂S, and UART.

Example code is supplied for users who want to develop custom applications in C, Visual Basic, LabVIEW, etc.

The Software Development Kit (SDK) that's included allows scope control using custom or third-party software. The SDK and PicoScope are Windows-compatible.

The included software for Windows harnesses the PC's



WEIRDSTUFF[®]
WAREHOUSE
Software, Computers, Electronics, Equipment, Doohickies

WE BUY/SELL EXCESS & OBSOLETE INVENTORIES
FREE COMPUTER AND ELECTRONIC RECYCLING
GIANT 10,000 SQ. FT. AS-IS SECTION

**384 W. Caribbean Dr.
Sunnyvale, CA 94089**
Mon-Sat: 9:30-6:00 Sun: 11:00-5:00
(408)743-5650 Store x324
WWW.WEIRDSTUFF.COM

**Do you know how many watts
(YOUR MONEY)**
are going down the drain from
"THE PHANTOM DRAW?"

The KILL A WATT
meter is the
best way to help
you determine
your actual
energy draw
in ON and OFF
home appliances.

To order call 1 800 783-4624
or online www.nutsvolts.com
\$29.95 plus S&H

www.Primecell.com
Battery rebuilding service

Dead Batteries? Don't toss them. Send them to us - our rebuilds are better than original specifications.

Tools
Hilti Skil Milwaukee Panasonic B&D DeWalt Makita All 2-36 Volts

Electronics
Bar Code Scanners Surveying Printers Laptops Photography

Radios
APELCO UNIDEN G.E. ICOM KENWOOD MOTOROLA MIDLAND MAXON YAESU ALINCO

Visit www.primecell.com for important details
24 Hr Secure recorder tel-fax (814) 623-7000
Quotes email: info@primecell.com
Cunard Assoc. Inc. 9343 US RT 220 Bedford PA 15522

NKC electronics

MDE8051 Trainer
by Digilent

NKCelectronics.com/MDE8051

Includes the MDE8051 training board, power supply, serial cable

Purchase Orders are accepted from Educational Institutions, US Government and Research Centers

ROBOT POWER[™] Extreme Motor Speed Control!

MegaMoto - Motor Control for Arduino[™]

- 6V-28V - 13A / 30A peak
- Single H-bridge or dual half
- Current and Temp protected
- Jumper select Enable and PWM pins
- Up to three units stack on one Uno/Duemilanove

Dalf - \$250

- Closed-loop control of two axes
- Windows GUI for all features
- Full PID position/velocity loop
- Trapezoidal path generator
- Giant Servo Model
- www.embeddedelectronics.net

>> Also H-bridges, gearmotors, plans, schematics <<

www.robotpower.com
Phone: 253-843-2504 • sales@robotpower.com

Quality Kits for All Your Electronic Needs!

Explore the World of Electronic Project Kits and Devices. *LOTS OF GREAT GIFT IDEAS!*

We offer a wide selection of electronic kits for beginners, as well as more advanced projects for students and experts! 1-888-GO-4-KITS

We also carry Arduino and Raspberry Pi products, as well as a variety of speed controllers, timers, power supplies, plus lots more!

QKITS.COM

SLIDE MOUNT for MOBILE RADIOS UNIVERSAL: FITS MOST 2 WAYS

Model ET-850 Universal Slide Mount. Standard mounting platform, specify radio make/model. Ends blind fumbling to release hidden fasteners. Easily transfer radios from vehicle to vehicle. Also use in the office, job trailer, marine too. Black powder coat finish. Volume prices.

(Ask about our maximum security Locking Radio Cage™ Mount)

E TIP, Inc., North Aurora, IL 60542 USA
(800) 530-5064 FAX (630) 801-9569
www.etipinc.com | sales@etipinc.com

E TIP is a Veteran Owned Small Business

Create a Custom Touchscreen in 4 Easy Steps.

1. Order a Development Kit (Starts at \$349)
2. Design to Your Specifications
3. Get Prototype Running in Days
4. Order Production Units (5+ Years Availability)

Learn more at ReachTech.com/NUTS/
510-770-1417
sales@reachtech.com

Your Microcontroller

Our Display Module

Free Shipping Code: NUTS

year warranty starting at \$260.

For more information, contact:
Saelig Company, Inc.
www.saelig.com

If you have a new product that you would like us to run in our New Products section, please email a short description (300-500 words) and a photo of your product to:

**newproducts
@nutsvolts.com**

In the Nuts & Volts Webstore NOW!

NUTS AND VOLTS

NV BOOK SPECIALS

309 Pages Only \$23.95

SAVE 20%!

224 Pages Only \$19.95

To order call 800 783-4624 or visit: <http://store.nutsvolts.com>

Limited time offer

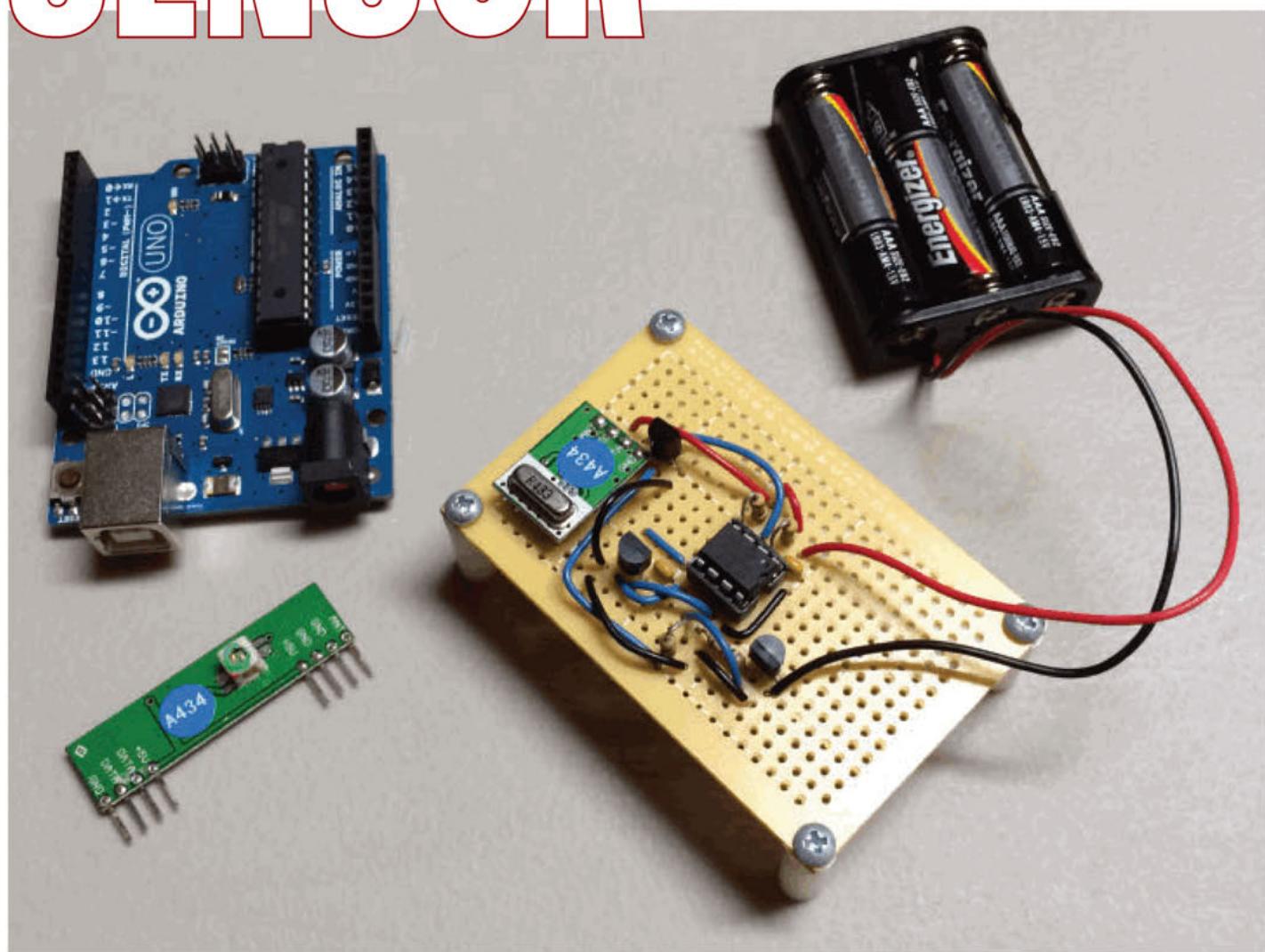
processing power, storage, graphics, and networking capabilities. The user interface is easy for novices to learn, but professional users will find many advanced features including spectrum analysis, persistence display, automatic measurements, advanced triggers, and channel math capabilities. Users can download software updates, feature extensions, and improvements free of charge.

The PicoScope 2000 series is supplied complete with two passive x1/x10 probes and a carry case. They are available now with a five

BUILD A PIC-BASED REMOTE TEMPERATURE SENSOR

By Ryan Clarke

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/january2014_Clarke.



The weather is a topic of interest to everyone — especially us nerd types (a point of pride). While there are plenty of remote weather stations available for purchase, they can be a bit on the pricey side. It would be a lot more fun to design and build your own weather station ... not to mention cheaper and educational! In this project, we will build a remote sensor which can be integrated into a homemade weather station using a digital microcontroller and an analog sensor for the most popular of all weather parameters: the temperature.



Introduction

A couple of years ago, I purchased a pair of RF receiver and transmitter links from SparkFun Electronics during their popular annual Free Day. Unfortunately, due to life commitments that always seem to get in the way of hobbies, they were thrown on the "I'll check that out later" pile. Fast forward to present day, where I was motivated to re-visit them and build a test circuit with a real world application. So, I settled on a remote temperature sensor. (You can never have too many remote sensors, right?)

The concept is relatively simple: Powered by batteries and using a microcontroller, periodically sample a temperature sensor and send the data through the RF transmitter to a base station. To make the project a little more interesting, I decided to use an analog temperature sensor which would require the use of an analog-to-digital converter (ADC).

Additionally, a switching transistor will be used to turn the transmitter on and off as needed. In short, not only do we get to program a microcontroller and work in the digital world, but maybe we'll learn some new tricks in the analog domain, as well ... the best of both worlds!

Component Selection

First, let's talk about the transmitter link. The particular item I purchased from SparkFun is the Holy Stone MO-SAWR-A (WRL-08949, replaced by WRL-10534). Transmitting on 433.92 MHz, this device operates between 1.5V-12V at 2.9-59 mA, outputting -8.5 dBm to 16 dBm (141 μ W to 40 mW) to a 50 Ω load. It has four pins: GND, Data In, VCC, and Antenna. The Data In pin accepts serial data at 2400-4800 bps which is modulated and transmitted via amplitude shift keying (ASK). We will be transmitting at 2400 bps since the receiver portion is limited to that data rate.

In order to maintain a reasonable RF power output – but also keep the circuit simple – I decided to use three AAA alkaline batteries. These will provide a nominal 4.5V and 1,000 mAh, down to about 2.7V when the batteries are fully discharged. According to the transmitter datasheet, we can expect approximately

11-22 mA of current draw and an RF output of 4-10 dBm (2.5-10 mW). However, in the lab I measured a significantly less current draw of 3.87-6.80 mA.

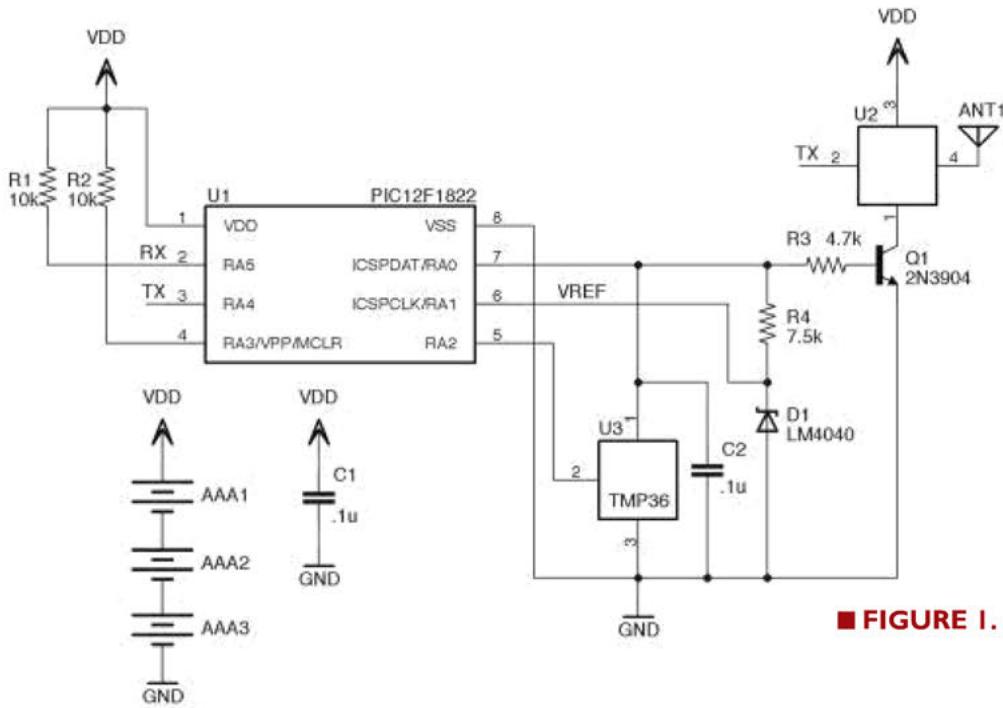
After considering the transmitter link, power source, and the desire to use an analog temperature sensor, I settled on the eight-bit, eight-pin PIC12F1822-I/P as the core of the system. This particular part has a programmable internal oscillator, an ADC, a serial UART, and operates over a wide voltage range (1.8V-5.5V).

For the temperature sensor, I chose the Analog Devices TMP36GT9Z low voltage temperature sensor which operates between 2.7V-5.5V, is specified for -40°C to +125°C, draws a maximum of 50 μ A, is calibrated directly in °C with a typical accuracy of $\pm 1^\circ\text{C}$, and comes in the familiar TO-92 package. At 25°C, the TMP36 outputs 750 mV and has a linear 10 mV/°C gradient.

To obtain accurate readings from an ADC, a consistent fixed voltage reference is required. A bonus of the PIC12F1822 is that it has its own programmable fixed voltage reference. Unfortunately, it has an accuracy of -8% to +6% – not exactly what we're looking for.

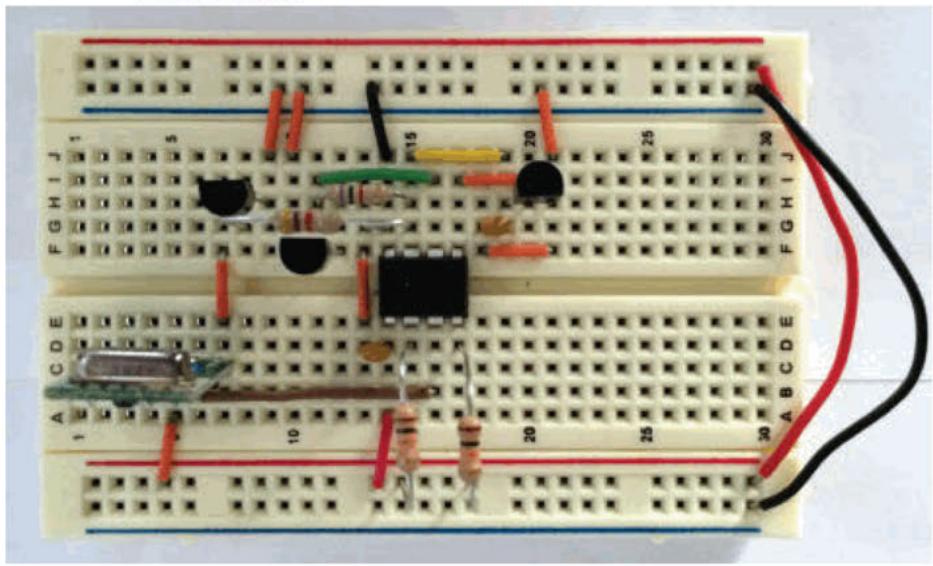
In order to support the full voltage output of the TMP36 (2.0V) and make the math on the raw ADC data easier (the PIC12F1822 has a 10-bit ADC), I chose the Texas Instruments LM4040C20I 2.048V precision micropower shunt voltage reference. The LM4040 is accurate to 0.5%, has low output noise, draws as low as 45 μ A, and is also available in the TO-92 package.

The switching transistor is simple and any general-purpose NPN bipolar junction transistor (BJT) will work. I decided on an old friend – the ubiquitous 2N3904 – suitable up to 200 mA in the TO-92 package.



■ FIGURE 1.

■ FIGURE 2.



A Deeper Look

Referencing the schematic (**Figure 1**) and the breadboard circuit (**Figure 2**), the PIC (U1) is relatively straightforward to wire up. A ceramic 0.1 μF capacitor (C1) is used across V_{DD} and GND to filter out any power supply transients and noise. Pin RA5 is connected to V_{DD} through a 10 k Ω resistor (R1) to prevent the pin from floating, thereby preventing excess power consumption. This pin will be assigned as the UART RX pin which will not be used in our design. Another 10 k Ω resistor (R2) is used to tie MCLR to V_{DD} . While this is technically unnecessary due to an internal weak pull-up in the PIC, it helps with noise immunity.

The transmitter link (U2) requires nothing special; just connect the appropriate pins as shown in the **schematic**. The Data In pin connects directly to pin RA4 (UART TX) of the PIC with no pull-up resistor required.

The LM4040 (D1) requires a shunt resistor (R4) to maintain the appropriate bias current while taking into consideration the load current which, in this case, is the PIC input leakage current. Referring to the example in the LM4040 datasheet, we can compute the minimum and maximum values for R4. It is already known that V_{DD} will be between 2.7V-4.5V and V_Z is 2.048V. The range for the load current (I_L) is 5-125 nA (from the PIC datasheet) and the range for the bias current (I_Z) is 75 μA to 15 mA. This yields a maximum resistance for R4 of 8,678 Ω and a minimum resistance of 164 Ω . In the interest of preserving the batteries (and more importantly, because I had it in my toolbox), I chose R4 to be 7.5 k Ω . This would set I_Z between 87 μA and 327 μA . The TMP36 (U3) is a no-brainer. A ceramic 0.1 μF capacitor (C2) was added between $+V_S$ and GND for noise immunity, per the recommendation of the datasheet.

Now it's time to exercise our knowledge of

transistors! To compute the base resistance (R3) required to saturate Q1 and thus use it as a switch, we must consider the case of the batteries being almost completely discharged, providing only 2.7V. This would bring the collector load (IC) to a minimum.

Referencing the **sidebar**, we can now compute the required maximum resistance for R3. Using the 2N3904 datasheet to find the value of V_{BE} when the device is saturated (approximately 0.75V), I compute R3 as 5039 Ω ... or 4.7 k Ω for a nearest standard value. This will guarantee the saturation of the transistor for the entire useable battery voltage range.

The Softer Side

Now it's time to write some code ... or at least it's time for me to explain the code that I wrote! Due to the fact that the PIC12F1822 is in the enhanced mid-range family of microcontrollers, I've elected to use the C language (using Microchip's free XC8 compiler) to write the software. This certainly makes coding easier, but keep in mind that using assembly on this device could possibly create faster leaner code. However, that's not a huge concern for this application.

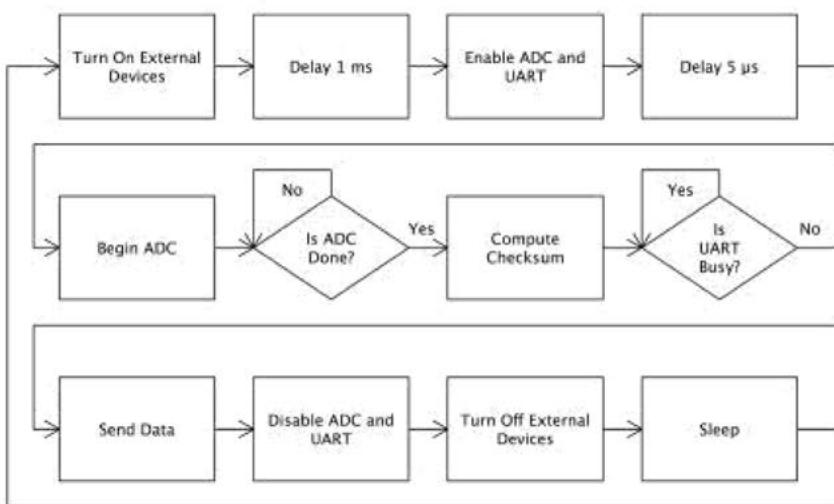
Referencing *temperature_transmitter.c*, the first order of business is configuring the PIC appropriately by setting the desired device configuration words:

- Internal/External Switch-Over (IESO) – disabled (two-speed start-up disabled)
- Oscillator (F_{OSC}) – internal
- Fail-safe Clock Monitor (FCMEN) – disabled (only useful with an external oscillator)
- PLL (PLLEN) – disabled
- Brown-Out Reset Voltage (BORV) – high (2.7V typical)
- Low Voltage Programming (LPV) – disabled

There are several more configuration words for this device, but I've left them at their defaults. Of note, we are using the brown-out reset capability of the PIC which will hold the device in reset when it drops below a set threshold; in this case, 2.7V. All configuration words can be referenced in the XC8 documentation (*pic_chipinfo.html*, supplied by Microchip).

Next up are some macros used in the source code. To save power (a faster clock equals more power consumption) but maintain a reasonable instruction execution clock ($F_{OSC}/4$), the oscillator is left at its reset value of 500 kHz. To reflect this is *_XTAL_FREQ*, which is required to be defined in order to use the delay functions

■ FIGURE 3.



provided by Microchip. The DEVICE_ID is the data address for the transmitter (to be used in the receiver for identification) and the PREAMBLE is used to condition the receiver's automatic gain control (AGC). More on this later.

With that done, the first thing we want to do is set the watchdog timer interval. You may now ask yourself, "What is a watchdog timer?" In the PIC (and other microcontrollers), a watchdog timer is an independent timer circuit that continuously runs in the background. If it is not periodically reset, the device will reset itself. This is useful in environments where the PIC may be subject to severe EMI and could lock up, or for poorly written code that locks itself up.

An added bonus for us is that it can be configured to run in the background while the device is asleep, and will subsequently wake up the device and continue execution when it reaches its programmed interval. This allows us to use it as a convenient power-saving method. For my purposes, I set the watchdog timer interval to 64 seconds (while testing, I lowered the interval to two seconds).

Before we can get to the good stuff, we still need to do some standard PIC boilerplate. Make sure to clear PORTA and set the appropriate pins as inputs and outputs with the TRISA register. Always clear the port first, as this will prevent a short circuit if a pin is erroneously config. After, we must enable the ADC for the appropriate pin (RA2) and disable the others; otherwise, they cannot be used as digital I/O.

Alright, time to set up the UART. A really nice feature of this particular PIC is the ability to reassign the TX/RX pins of the UART. The default is RA0/RA1, but I would like to use RA4/RA5. The baud rate generator needs to be set up for 2400 bps, eight-bit, high-speed, asynchronous mode, and the value for it can be computed with the following equation:

$$SPBRG = \frac{F_{OSC}}{(\text{desired baud rate} * 16)} - 1$$

With our requirements, this results in a SPBRG value of 12. Out of curiosity, we can find our true baud rate and error rate by reorganizing the equation which results in 2403.8 bps and 0.16%, respectively. Finally, the UART must be enabled, completing the setup process. The ADC setup can be a bit tricky and requires some study of the datasheet to understand. First, we must set the conversion clock. To do this, we must first know our minimum conversion time (T_{AD}) which is 1 μ s per the datasheet. With a clock frequency of 500 kHz, this results in an instruction cycle period of 2 μ s. If we select a conversion clock of $F_{OSC}/2$, we will be right at the minimum.

The positive voltage reference (V_{ref+}) must be set to an external V_{ref+} (the LM4040). Next, set the appropriate analog channel used for the ADC (AN2) and then set the conversion data to be right-justified.

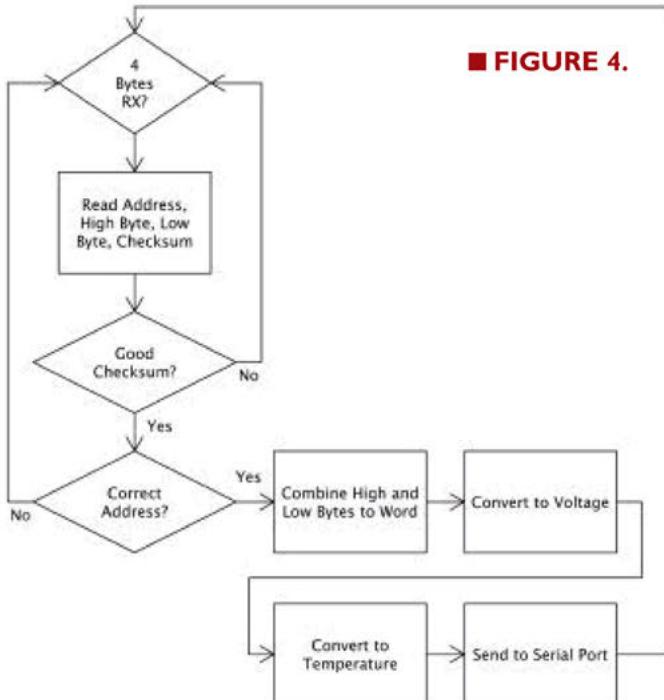
Referencing Figure 3, we get a picture of the main program loop. The first order of business is to turn on our external devices and then delay for at least 1 ms to allow for them to stabilize (specifically the TMP36). Following this, the ADC and UART TX must be enabled.

Okay, let's convert some voltage! To use the ADC in the PIC, we must first delay for the minimum acquisition time (T_{ACQ}). The typical value on the datasheet is 5.0 μ s. There is also an equation and example published in the datasheet used to compute the minimum T_{ACQ} , but for now we'll stick with 5.0 μ s. Once the delay is over, start a conversion by setting the ADGO bit and wait until the conversion is complete.

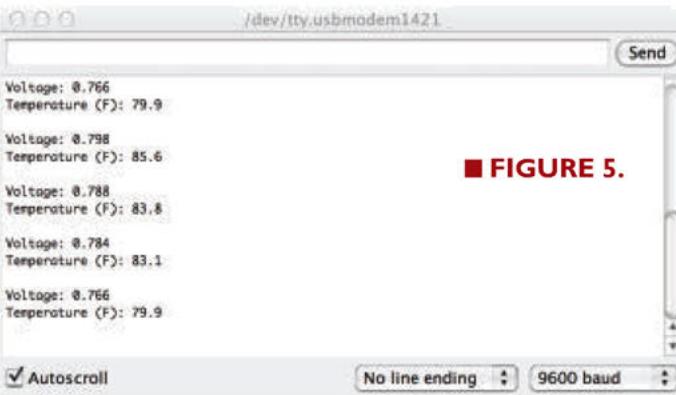
Well, that was easy! Now, let's compute a checksum and send some data. The checksum is a simple modular sum which is computed by summing all the data (DEVICE_ID and ADC result) together as an unsigned byte (discarding overflow), and performing a two's complement on the result. This will help us error-check on the receiver side. Next, we must send the PREAMBLE to the receiver in order to de-gain its AGC. The receiver link will gain-up the AGC when it is not receiving a strong signal, which results in a lot of noise and bad data.

In order to synchronize the transmitter and receiver and de-gain the AGC, a preamble of alternating bits is sent before the data. I suggest sending at least two bytes' worth of preamble to ensure success. After the preamble, we'll send the device ID, the high byte of the ADC result, the low byte, and the checksum. Before each byte, check to make sure the UART is not currently sending data. Also, after the final byte, verify that the UART has sent all the data.

Time to go to sleep! Disable the UART TX and ADC and switch off the external devices. Finally, put the PIC to



■ FIGURE 4.



■ FIGURE 5.

initialized in the `setup()` function. The serial port is set to 9600 bps and the SoftwareSerial port to 2400 bps. The main program loop (**Figure 4**) consists of reading the data, verifying it, converting it to temperature, and displaying the data. First, wait until at least four bytes of data are available in the SoftwareSerial port RX buffer. Once it is, read four bytes in the expected order: transmitter address, high data byte, low data byte, and checksum.

To verify the checksum, we simply sum all four bytes as an unsigned byte (discarding overflow) – the total of which should be zero. Next, check the transmitter address; if it's the one we are looking for, then convert the raw data to temperature and send it to the computer. To do so, first combine the high and low byte into one 16-bit word and convert it to a voltage:

```
unsigned short data = (high_byte << 8) | low_byte;
float voltage = (data / 1024.0) * 2.048;
```

Next, convert the voltage to a temperature. Remember, the TMP36 will output 750 mV at 25°C and has a 10 mV/°C gradient:

```
float temperature_c = ((voltage - 0.750) * 100)
+ 25;
float temperature_f = (1.8 * temperature_c)
+ 32;
```

Pretty easy. Finally, send it to the computer with some `Serial.print()` function calls and watch the data populate on the serial monitor.

Let's Go for a Test Ride!

With the transmitter in one room and the receiver in another, I had little issue receiving steady data. At times, the receiver would miss a transmission but that's not entirely surprising considering the specifications of the receiver link and the fact that the ISM band this operates on is very busy.

To verify the temperature sensor is indeed working, I put my finger on the TMP36 and watched the temperature increase and then decrease when I removed my finger (**Figure 5**). The current draw when the PIC is asleep and the devices are off is between 22.2-25.4 µA. When transmitting, it increases to 4.80-8.56 mA. Even with the small AAA batteries, you should be able to get about 39,000 hours out of them. Not bad!

There is Room for Improvement

There are a few other ways to skin this cat which could improve the design. First, we could use a digital temperature sensor in the transmitter. This would cut down on the number of parts required, but could

sleep and we're done. Wash, rinse, and repeat.

Anyone Out There?

For the base station, using the Holy Stone MO-RXLC-A (WRL-08949, replaced by WRL-10532), I decided to go with an Arduino Uno with the RF receiver link wired per the datasheet. I used the included SoftwareSerial library to receive the data from the receiver link at 2400 bps, process it in the Arduino, and then send the data to a computer over the built-in serial/USB interface to ultimately be displayed on the serial monitor included in the Arduino IDE (Integrated Development Environment).

Looking at `temperature_receiver.ino`, we first must create a SoftwareSerial object using pin 10 for RX and pin 11 for TX (we will not actually be using the TX portion). Then, both the serial and SoftwareSerial ports must be

complicate the coding a little, depending on the interface (1-wire, I²C, SPI, etc.). The RF transmitter and receiver links could be upgraded to devices that implement FSK versus ASK, which would provide for more reliable data reception. You could also power the transmitter with a 9V battery which would provide more RF power output.

Another option is to use a microcontroller with a built-in RF transmitter, such as the rfPIC line of devices. The challenge there would be the surface-mount package. Regardless, the circuit is small enough to fit in a very small enclosure. Add an external antenna to both the transmitter and receiver and it will increase the range of the system considerably. A follow-on project would be to build a receiver around another microcontroller and an LCD display.

I hope you enjoyed this project. There is plenty of room for improvement and expansion. Use your imagination, dig through some datasheets, and see what you can come up with. **NV**

R1	10K	10KQBK-ND
R2	10K	10KQBK-ND
R3	4.7K	4.7KQBK-ND
R4	7.5K	7.5KQBK-ND
C1	0.1 μ F	399-4266-ND
C2	0.1 μ F	399-4266-ND
D1	LM4040C20I	296-21579-1-ND
Q1	2N3904	2N3904FS-ND
U1	PIC12F1822-I/P	PIC12F1822-I/P-ND
U2	TWS-BS	WRL-10534 (SparkFun.com)
U3	TMP36GT9Z	TMP36GT9Z-ND
U4	RWS-371	WRL-10532 (SparkFun)

PARTS LIST

Note: All resistors are 5%. All part numbers are from [DigiKey.com](#) unless otherwise noted.

Saturating an NPN Bipolar Junction Transistor

Transistors have the wonderful ability to be used as both an analog device (e.g., an amplifier) and a digital device (e.g., a switch). To use a transistor as a digital switch, we must force it into the saturation region. This is actually quite simple. We know that the collector current (I_C) and base current (I_B) are related to each other as follows:

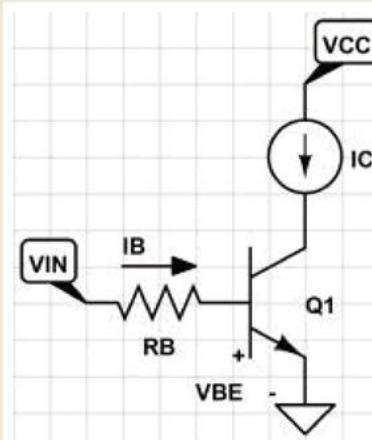
$$I_C = \beta I_B$$

where β is the common-emitter current gain, also represented as h_{FE} .

Typically, this is on the order of 100 for most general-purpose transistors biased in the active region. However, if you look at the datasheet of the very common 2N3904, for example, and pay attention to parameters for collector-emitter saturation voltage or base-emitter saturation voltage, you'll notice β in those cases is equal to about 10. This is a typical value used for saturating a transistor. Another parameter to reference (which was just previously mentioned) is the base-emitter saturation voltage ($V_{BE(sat)}$). Armed with these two values, we can easily compute the maximum base resistance (R_B) useable to saturate a transistor:

$$V_{IN} - I_B R_B - V_{BE(sat)} = 0$$

Simply solve for R_B !



Imagine this...

- a small and inexpensive
- USB or ethernet device
- with a rich set of features
- including PLC functionality and 3-axis stepper motor controller
- all accessible via free .NET, ActiveX or C++ library
- cross-platform
- configurable with free software

PoKeys



Or this...

- Smallest USB 2.0 portable 1MS/s oscilloscope
- Data acquisition of analog and digital signals
- Data recording
- Export to CSV, XLS, PDF and HTML
- Simple usage of advanced features
- Examples for C++, VB, Delphi and LabView
- Free software and updates

PoScope Mega1+



PoLabs
Visit us at
www.poscope.com



THE (NEARLY) UNIVERSAL PROCESS MONITOR

By Daniel Gravatt
dgravatt@juno.com

Post comments on this article and
find any associated files and/or
downloads at [www.nutsvolts.com
/index.php?/magazine/article/
january2014_Gravatt.](http://www.nutsvolts.com/index.php?/magazine/article/january2014_Gravatt)



I'm continually amazed by how much functionality microcontroller manufacturers can pack into their products. In even the most basic units available now, things that used to be considered "peripherals" such as non-volatile memory, timers, comparators, analog-to-digital converters, real time clocks, and all sorts of I/O ports are now built in. A complex system can consist of little more circuitry than the micro itself.

■ FIGURE 1. Completed display unit on the dashboard of my electric truck.



When I set out to build a display for the system parameters in my electric truck conversion, the PIC16F819's peripherals made for a very compact and inexpensive dashboard-mounted unit (**Figure 1**). I realized this was a very specific application for what could be a simple general-purpose process monitor and display for analog and pulsed signals.

Potential applications include alternative energy systems such as solar photovoltaic or thermal, wind turbines, or geothermal; backup power systems for your home or RV; keeping tabs on your aquarium or fish pond water quality; or maybe just building your own weather station from scratch.

This process monitor can accept three analog signals either as continuously variable voltages or pulse-width modulated signals, as well as one digital pulse input for a frequency counter. The software includes a conversion formula for each input with user-defined variables to turn raw signal data into scaled numeric values.

The software also accepts user-defined names for each value, and displays the four values and their names on a 2x20 (or larger) character LCD in real time. The user-defined variables and names are entered with a simple three-button interface and are retained in non-volatile memory.

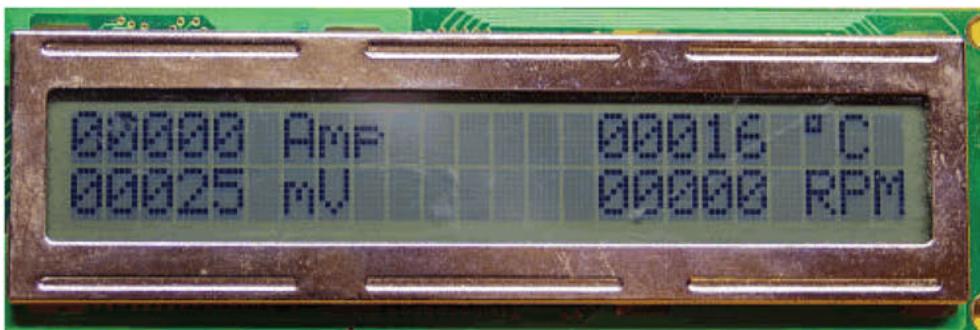
Software

The three analog signal inputs are converted to raw data by the PIC's built-in analog-to-digital converter (ADC) at 10-bit resolution, representing a range of zero to five volts (assuming a five volt supply voltage to the PIC). The conversion formula to turn this raw ADC data into scaled values is in the form:

$$\text{Scaled value} = ((\text{raw data value from ADC} * \text{variable1}) / \text{variable2}) + \text{variable3}$$

Variables 1 and 2 allow the raw data to be scaled to represent larger or smaller voltage ranges than five volts if you are using a voltage divider on the ADC input. This is what I had to do in my application, where the electric motor controller outputs were 12 volt signals. If you don't need to use one or both of these variables to scale your data, they can be set to one.

Variable 3 allows offsetting the scaled value to allow measuring and displaying



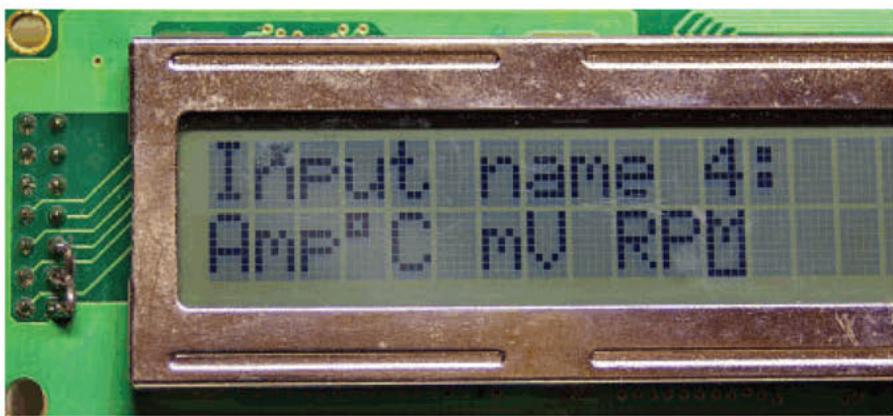
■ FIGURE 2. Operating display after names and variables are entered.

small changes in a large voltage, for example. If no offset is needed, this variable can be set to zero.

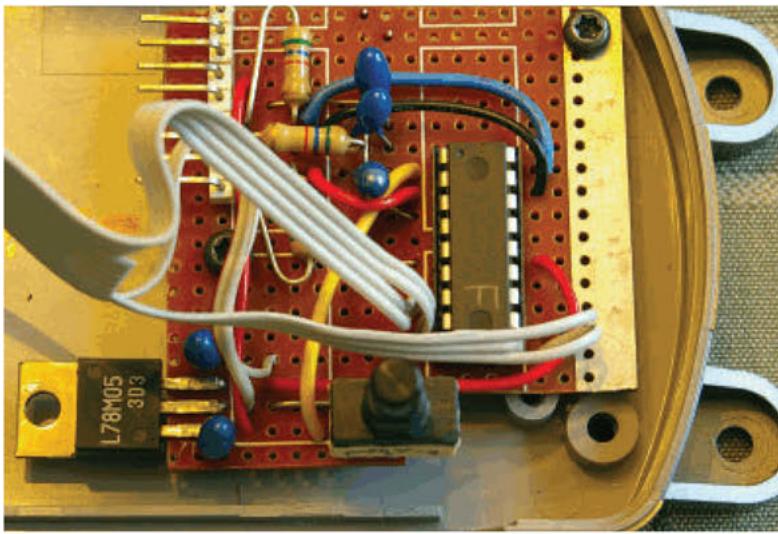
The scaled values are calculated as word-sized integers. While a lack of a decimal point appears to be a significant drawback, the formula above also allows us to work around this limitation in many cases. If you need to measure zero to five volts with a resolution of 10 mV, for example, adjust the variables accordingly and change the name of this reading from volts to millivolts (**Figure 2**).

When the process monitor is powered up, it will default to using the last set of variables and input names that were saved. To change these, hold the enter button as prompted on the start-up display. Each name can be up to three characters which are selected with the up and down buttons. Any ASCII character can be entered, including blank spaces if you don't need all three characters. Input 1 corresponds to the PIC's ADC channel 0; input 2 to ADC channel 1; input 3 to ADC channel 3 (not a misprint!); and input 4 to the timer1 counter/timer input on portB.6.

This somewhat odd numbering scheme results from the limited set of choices available for configuring the PIC's portA pins as digital or analog; since we need some of both, the best choice available gives us three analog and five digital pins on portA. Due to a limitation of the method the software uses to store the input names (solving one problem with EEPROM writes but creating a



■ FIGURE 3. Display showing entry of the fourth input name; previous names are also visible.



■ FIGURE 5. Printed circuit board for this project.

minor inconvenience in the process), the characters will all be displayed together as they are being entered (**Figure 3**).

Once the four input names are entered, the variables for each formula are set. Each variable is a word-sized numeric value, entered as individual digits with leading zeros as needed. Variables 1, 2, and 3 will be used in the conversion formula for input 1/ADC channel 0; variables 4, 5, and 6 for input 2/ADC 1; and variables 7, 8, and 9 for input 3/ADC 3.

The conversion formula for the counter/timer input on portB.6 differs slightly from the ADC conversion

formula in that there is no offset variable:

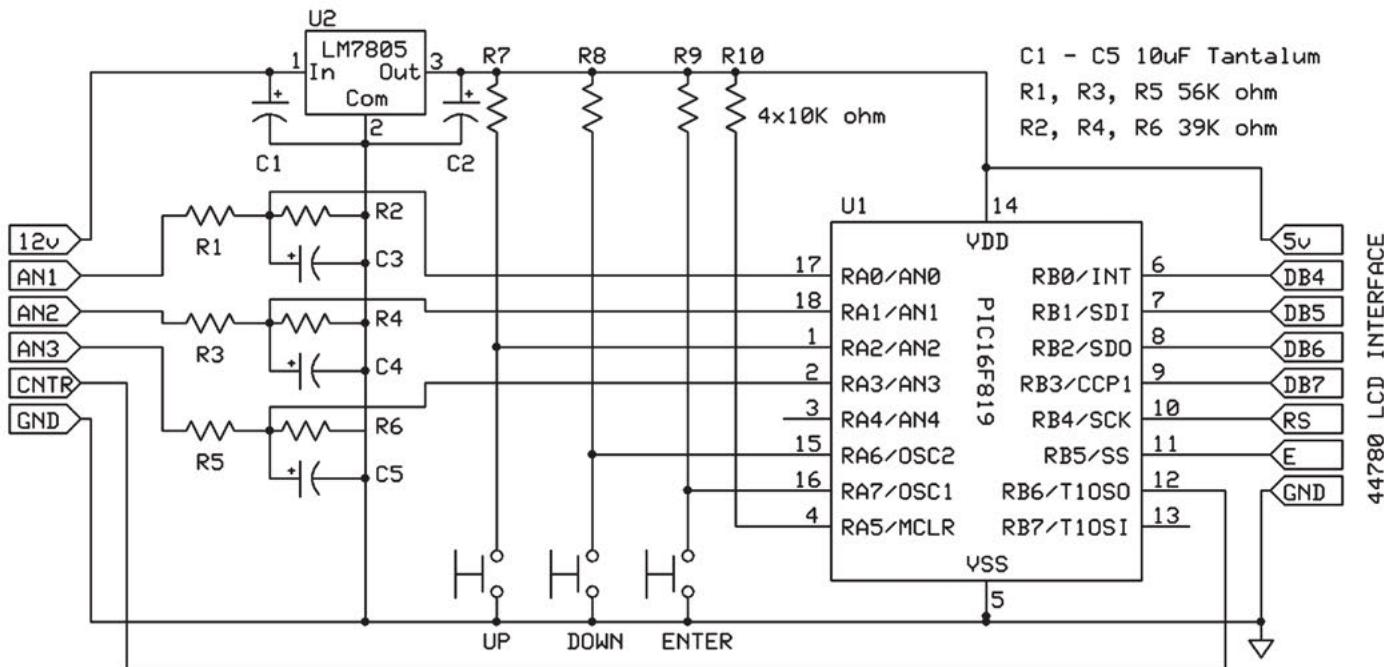
$$\text{Count} = (\text{raw count value from timer1} * \text{variable10}) / \text{variable11}$$

After all the names and formula variables are entered, the software resets and starts calculating and displaying your data with updates every half second. In **Figure 1**, you can see that my project's software and display format is slightly different from that described here because it's a dedicated unit that I don't plan to use for other purposes. If you have a similar permanent application in mind, you can omit the user interface buttons and input subroutines, and just hard-code the input names and variable values. The software described here (available at the article link) uses up just over 1.3K words of the PIC's 2K word Flash memory.

Hardware

The very simple schematic for the process monitor is shown in **Figure 4**. The three-button user interface uses the remaining digital-mode inputs on portA, while portB is configured with one digital input for timer1 on portB.6 and digital outputs to drive a HD44780-compatible alphanumeric LCD. One portA and one portB outputs are available in case you want to modify the code so your process monitor can also be a process controller.

The signal conditioning circuit for each analog input will have to be tailored to your particular application. It



■ FIGURE 4. Schematic of the process monitor. Refer to the text for analog input circuit options.

PARTS LIST

could be as simple as a direct connection from the sensor output to the ADC input if the sensor voltage is limited to less than the supply voltage of the PIC. Consult a qualified electrician before attempting to design a signal conditioning circuit for any high voltage signals!

The configuration shown in the **schematic** is what I've used in my electric truck application where the inputs are 12 volt PWM at a frequency of 60 Hz. The resistor voltage divider drops this down to five volts, and the parallel 10 μ F capacitor smooths the PWM into a constant voltage.

I have tested the timer1 input up to 100 kHz with a 50% duty cycle square wave with accurate results. If you modify the software, you may have to adjust the length of the PAUSE statement in the main subroutine to maintain the accuracy of your count, as this length takes into account the latency of the code in this loop to set a precise half second interval between displayed values.

The PIC16F819 has the option of using an internal oscillator and internal MCLR pull-up resistor. In this design, I'm using the internal oscillator, but have MCLR pulled up externally so the PIC can be easily hard-reset if desired. When you program your PIC, you will need to select the "INTIO2" oscillator configuration option to enable the internal oscillator. Note that not all programmer software uses Microchip's standard name for this option; my software calls it "IntRC."

Construction style is not critical unless your sensor outputs require special shielding, so a small piece of perfboard and a suitable plastic enclosure is probably all you will need for the build (**Figure 5**).

Now, dream up a project that needs real time process monitoring, build it, and then build this process monitor to help you keep tabs on it.

NV

ITEM

PIC16F819 microcontroller, 18-DIP
 (3) PC-mount pushbutton momentary switches
 (4) 10K ohm 1/4 watt resistor
 (2) 10 μ F 25V tantalum capacitor
 7805 voltage regulator, TO-220
 2x20 character LCD with HD44780 controller

DIGI-KEY PART

PIC16F819-I/P-ND
 450-1665-ND
 10KQBK-ND
 478-5813-ND
 296-13996-5-ND
 MOP-AL202C-BYFY-25E-3IN-ND

Parts List excludes input signal conditioning circuits.



www.allelectronics.com
Order Toll Free 1-800-826-5432

24VDC DPDT 25A RELAY Tyco P&B PRD-11DY0-24. 24 Vdc, 288 ohm coil. Contacts rated 25A, 240 Vac. 3.375" x 2.51" x 2.50" high. Screw terminals. CAT# RLY-2425	2-COND. POWER CORD 6' black SVT round cable. 0.23" nom. o.d. Molded strain relief. UL, CSA. CAT# LCAC-419
 \$7.95 each	 \$2.00 each
10 for \$1.85 each • 150 for \$1.75 each	
2 AMP CIRCUIT BREAKER P&B W28-XQ1A-2. Single pole, panel mount circuit breaker. UL, CSA. CAT# CB-2802	12VDC 750MA POWER SUPPLY Motorola MT20-21120-A04F 5.5' cable with 2.5mm coax plug, center+. RoHS, cULus. CAT# PS-12752
 \$1.75 each	 \$6.75 each
10 for \$6.50 each	
ENCLOSURE W/ MEMBRANE PUSHBUTTONS & BATTERY COMPARTMENT Hand-held enclosure designed for a medical device, 5.75" x 3" x 1.5". Six membrane push- buttons are built into the front cover with flexi-ribbon cable termination. Two 0.45" square cut-outs on either side of case. Features a battery compartment, folding desk-stand, a belt clip and a clear plastic window. CAT# MB-910	4-POSITION PC MOUNT TERMINAL STRIP Screw-clamp terminals on 5mm centers. Accepts up to 14 AWG wire. Rated 10A, 300V. Numbered 1-4. Consists of 2 interlocking modules that can be separated or added to other modules. CAT# TER-404
 \$2.00 each	 60¢ each
100 for 40¢ each	
RGB LED, 5MM 4-lead, red-green-blue LED. One common lead (anode) and one lead for each color. 5mm (T1 3/4). Water-clear in off-state. CAT# LED-258	PANEL METER, FULL SCALE-1MA 2.4" square panel meter. Full scale = 1mA DC. Imprinted with 0 - 15 and 0 - 3 VOLTS scales in black with white background. CAT# PM-376
75¢ each	 \$3.50 each
10 for 70¢ each 100 for 60¢ each	

All About Clocks for PICs

By Thomas Henry
studs.kreitzer@gmail.com

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/january2014_Henry.

All microcontrollers require a clock or oscillator to guide a program through its paces. It's the duty of such a module to indicate when an instruction should be fetched from program memory, decoded, and acted upon. Actually, even the simplest instruction is made up of a number of operations that must be sequenced in just the right order and at just the right moment. The clock, then, is like the conductor of an orchestra, coordinating all of the parts that make up the whole.

Since the clock is so important, PIC microcontrollers offer a broad range of options to choose from for your own applications. You can have clocks that run rapidly when a lot has to happen in a short span of time, or ones that consume negligible power for battery-operated rigs. Maybe synchronizing the PIC to a stable real time clock is important to you. In fact, there are eight or more clock options for the typical PIC.

However, if you grab the datasheet and try to sort out all the possibilities, you'll quickly come to rue that old phrase, "Can't see the forest for the trees." Datasheets for PICs run in the hundreds of pages and aren't necessarily organized for best learning. And, of course, the occasional error works its way in, making it even harder for the newcomer to get up to speed.

As someone who has spent his entire adult life in teaching, I've always intuitively valued the importance of a well organized presentation. You begin with the

"forest" and only approach the "trees" afterwards. That's what we'll do here — getting the big picture in mind first and then tackling the details when they're finally needed.

So, if you've been dismayed in the past at how complicated the clock options for PICs seem to be, tag along now and see how the right approach can make all the difference. To turn this into an active learning experience, we'll conclude with some actual experiments you can conduct on the breadboard. By the end of our session together, you should be all set to start using PIC clocks with confidence.

Primary Clocks

To keep things concrete, I'll focus on the PIC16F88 which is one of the most popular of all microcontrollers among DIYers. However, other PICs will sport many of the same options — even the smaller eight-pin chips.

The first concept to nail down is that the PIC can run either on a primary clock or a secondary clock. The former accounts for most situations. In particular, a typical project contains a single clock guiding the microcontroller along and that's the end of the matter. In more advanced applications, we may want a secondary clock to kick in whenever the primary clock is out of the scene. There are several reasons for desiring such redundancy, and we'll explore them in just a bit.

For now, let's focus on the primary clock which is all we need for most projects. Refer to **Figure 1** which shows what's available. This tree diagram organizes the types of primary clocks available within the PIC16F88 in a top-down fashion. To begin, look at the first branches of the tree diagram. A clock can be an internal RC affair or, if you prefer, you can supply your own resistor and capacitor and end up with an external RC arrangement. For situations that demand more precision, an external crystal or ceramic resonator can be employed. Lastly, if you'd like, you can drive the PIC with an existing external clock.

Adding a bit more detail now, the branches of the tree split again showing the eight individual modes. Each has an abbreviated name given to it by Microchip. These are worth learning now to simplify reading the datasheet later on. Let's scan these specific modes from left to right. The first is INTRC which stands for internal RC oscillator. Running at an approximate 31 kHz, this is a low speed affair built inside the PIC. While that's pretty slow by any standards, it does have the advantage of being simple and consuming very little current.

The next mode is INTOSC. Like INTRC this is internal (requiring no outboard components), but can run at seven different speeds all the way up to 8 MHz. In some of the PICs, INTRC and INTOSC are truly independent, but not so in the PIC16F88 we're examining. For this reason, in the datasheet the writers sometimes use INTRC to mean INTRC only and at other times to signify either INTRC or INTOSC. That only adds to the confusion, so I'm going to keep the names separate here.

In either of these modes, port line A.6 can be configured to follow the clock divided by four if desired. This is symbolized $\varphi/4$ – read "phi divided by four." Use this output if you'd like to synchronize some external gear to the micro. The datasheet calls this mode INTIO1, if you're curious. Otherwise, you can optionally free up pin A.6 for ordinary digital I/O, referred to as INTIO2 mode.

Moving to the external RC modes, the first is simply called RC. In this case, an external resistor and capacitor on pin A.7 get things going. Assuming a +5V power

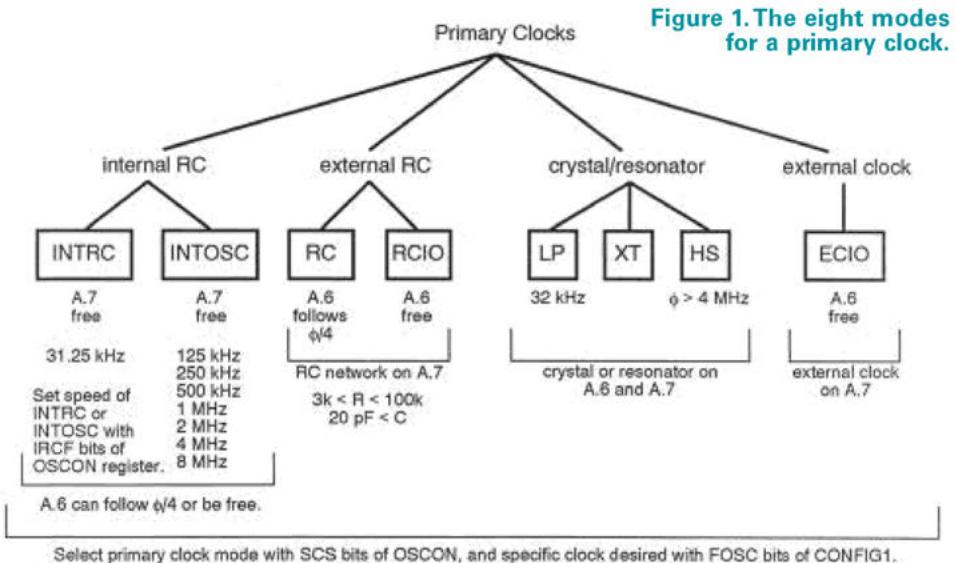


Figure 1. The eight modes for a primary clock.

supply, the resistor should lie between 3K and 100K, while the cap should be greater than 20 pF. With the smallest values, the clock will max out at around 4 MHz. Like we saw with the internal clocks – if desired – A.6 can be caused to follow $\varphi/4$ (that's RC mode) or else be free for ordinary use (that's RCIO). Incidentally, the resistor could be a potentiometer wired as a rheostat which would give you a variable clock.

Crystals and ceramic resonators are attractive since they're much more accurate both in the short term and long term, and also with respect to tolerances and temperature. There are three modes here. The first, LP – which stands for low power – is meant to be used with a 32.768 kHz tuning fork type crystal. These pop up commonly in wristwatches and real time clocks. The crystal and associated load capacitors connect to pins A.6 and A.7. The next mode – called XT – is similar but meant for use with higher frequency crystals and resonators up to about 4 MHz. The last mode is HS – standing for high speed – and is needed for crystals or resonators cranking above 4 MHz. Just so you know, the upper limit for the PIC16F88 is 20 MHz. What's neat about these three modes is that the chip automatically selects the correct gain of the internal driver to the crystal based on whether you specify LP, XT, or HS.

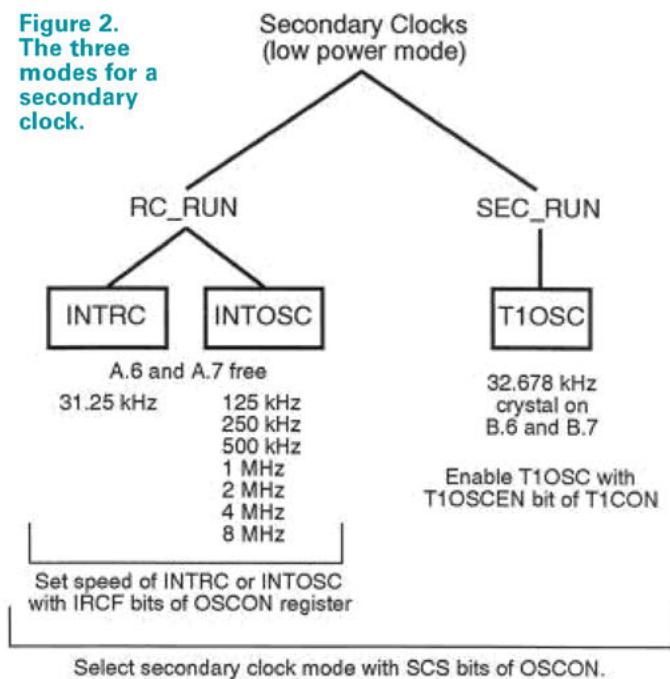
The last mode is ECIO which simply lets you pump in an external clock signal on pin A.7. This could come from any sort of clock circuit that puts out square pulses swinging from ground to the supply voltage.

Peeking ahead just a bit to the experiments, **Figure 6** shows the hardware arrangements involved. We'll see how to stipulate any of these eight modes in just a moment, but first let's take a look at the secondary clock options.

Secondary Clocks

Figure 2 shows a branching tree diagram for the available secondary clocks. Recall that any of these can

Figure 2.
The three modes for a secondary clock.



take over duties from the primary clock under certain circumstances. The datasheet likes to group these into the categories RC_RUN and SEC_RUN, but both indicate secondary clocking.

There are two choices for RC_RUN. Either the INTRC or INTOSC internal clocks described earlier can be used; the only difference really being the rate of oscillation.

Or, if desired, you can attach the 32.768 kHz crystal and load capacitors to B.6 and B.7. What's so cool about this is that the watch crystal always runs – even when you put the chip into what's known as sleep mode. This mode – designated T1OSC – is ideal for real time clock applications. (Normally, the PIC will shut down unused clocks when going to sleep).

Observe that primary clocks use A.6 and A.7 in various combinations, but secondary clocks do not – at least for the PIC16F88. The schematic in **Figure 7** (for one of the experiments coming up) shows what's involved.

With that, we've concluded our overview of the primary and secondary clocks. Why not take a few moments to study **Figures 1** and **2** once more to really fix the distinctions and details in your mind before proceeding.

How to Make Your Selection

So, you've decided what kind of clock you want. Now, how do you express your wishes to the PIC? The answer lies in the configuration bits and three special registers. The configuration bits are set during the burning (Flashing or programming) phase, while the register bits are accessible during runtime. Here's the scoop.

Figure 3 illustrates the two sets of configuration bits within the PIC16F88. (Simpler PICs only have one set.) Yes, there's a lot here, but when it comes to setting up the clock only a handful need concern us. For example, the three bits labeled FOSC designate the primary oscillator as described earlier. In other words, these establish the default behavior on power-up.

In the second set of configuration bits, you'll find two that govern what will happen when you switch over from the primary to secondary clocks, or vice-versa. We'll save that for the next section.

Once a program is running, you can manipulate the clock (or clocks) in various ways by modifying the registers OSCON, OSCTUNE, and T1CON. These are shown in **Figure 4**. Let's ponder the details.

Perhaps the most important bits in OSCON are the ones labeled IRCF, which stands for internal RC oscillator frequency. As the name suggests, these bits select the desired clock rate of the internal oscillator(s). Next up are the two bits labeled SCS, denoting system clock select. Here, you can choose to use the primary or secondary clocks. It's no big surprise that the register OSCTUNE lets you fine-tune the INTRC or INTOSC clock frequency. (On some PICs, only INTOSC is affected.) A six-bit two's

CONFIG1						
CP	CCPMX	DEBUG	WRT1	WRT0	CPD	LVP
bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7
BOREN	MCLRE	FOSC2	PWRTE	WDTEN	FOSC1	FOSC0
bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

FOSC2 : FOSC1 : FOSC0

111	RC
110	RCIO
101	INTRC or INTOSC, A.6 follows $\phi/4$
100	INTRC or INTOSC, A.6 free
011	ECIO
010	HS
001	XT
000	LP

The FOSC bits are used to set the primary oscillator mode.

CONFIG2						
n/a	n/a	n/a	n/a	n/a	n/a	n/a
bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7
n/a	n/a	n/a	n/a	n/a	IESO	FCMEN

bit 6 bit 5 bit 4 bit 3 bit 2 bit 1 bit 0

IESO = Internal/External Switchover
FCMEN = Failsafe Clock Monitor Enable

IESO is used to enable the Two-Speed Clock startup mode. It is needed only when the primary clock is LP, XT or HS. After a reset, the clock will be INTRC briefly, switching over to the primary source once the crystal has warmed up.

If FCMEN is set and the clocking from LP, XT, HS or T1OSC fails, then operation will automatically continue with INTRC or INTOSC instead.

Figure 3. Configuration bits are set when the PIC is burned.

complement number is all it takes. Negative numbers slow the clock down, while positive ones speed it up. Tuning over a range of $\pm 12.5\%$ is possible.

The register T1CON concerns itself with controlling Timer 1 which is typically clocked by an external 32.768 kHz crystal which can also do double-duty as the system clock. You might remember this as T1OSC mode from **Figure 2**. The important flag here is T1OSCEN which enables the Timer 1 crystal oscillator.

At this point, we've hit the fundamentals of clocking the PIC for most common situations. Let's finish up by taking a brief look at the secondary clocks and what they're good for.

Clock Switching

The day may come when you'll want to go beyond just turning on a primary clock and letting it do its thing. If so, then you'll need to know a little bit about clock switching. **Figure 5** gives the big picture. Essentially, you can manually switch between primary and secondary clocks, or do so automatically under certain conditions.

To manually switch from one to another, you're back to **Figure 4** and simply need to alter a few bits in the three registers described there. For example, to change from INTRC to INTOSC, you'd adjust bits IRCF0 through IRCF2 as required. Or, to go from a crystal oscillator, say, to INTOSC, head to SCS0 and SCS1 which let you flip back and forth from primary to secondary. You get the idea; manually switching oscillators is just a matter of bit-twiddling in your program.

So, what about automatic switchovers? Well, there are two situations in which it's handy to have one clock take over for another. The first is the failsafe mechanism. Imagine you're designing a circuit to operate in a critical environment in medicine or industrial safety monitoring, for example. If the primary oscillator was a crystal, for instance, and it failed for some reason, then the PIC would simply come to a standstill. Not good! With a secondary clock in the wings, the chip could switch over to it almost seamlessly and still keep operating until the original fault could be attended to. As **Figure 5** shows, the failsafe feature can check for problems with any of the four crystal modes (LP, XT, HS, or T1OSC). It's enabled via the FCMEN bit in CONFIG2.

If avoiding delays is a concern, then the two-speed wake-up option might prove useful. Here's the basic idea. Suppose you were operating the circuit on a crystal. As you probably know, crystal oscillators take a moment or two to warm up. The PIC understands this and patiently waits a bit before letting the clock start shooting the instructions through the pipeline. In other words, your project is just sitting there doing nothing for a few moments. On the other hand, the internal oscillators INTRC and INTOSC are essentially put in gear instantaneously. So, why not choose one of these to serve as a secondary clock?

OSCON							
n/a	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

IRCF = INTRC or INTOSC Frequency
OSTS = Oscillator Start-Up Time-Out Status
IOFS = INTOSC Frequency Stable
SCS = System Clock Select

IRCF2 : IRCF1 : IRCF0			SCS1 : SCS0		
000	31.25 kHz	INTRC	00	Primary Oscillator	
001	125 kHz		01	Secondary is T1OSC	
010	250 kHz		10	Secondary is INTRC or INTOSC	
011	500 kHz		11	n/a	
100	1 MHz				
101	2 MHz				
110	4 MHz				
111	8 MHz				

OSCTUNE							
n/a	n/a	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Fine tuning for INTRC and INTOSC covers a range of $\pm 12.5\%$.

100000 minimum
000000 center frequency
011111 maximum } two's complement number

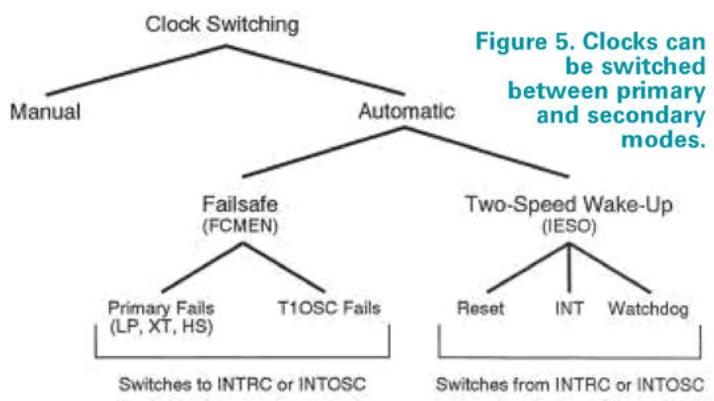
T1CON							
n/a	T1RUN	T1-CKPS1	T1-CKPS0	T1-OSCEN	T1-SYNC	TMR1CS	TMR1ON
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

T1OSCEN is the Timer1 Oscillator Enable bit and must be set to start the oscillator. T1RUN is a read-only flag and will be set when the system clock is derived from T1OSC.

Figure 4. These registers can be changed during runtime.

Now, the sequence will be that the internal oscillator handles everything (possibly at a slower rate, but at least it's moving!) until the crystal is ready to assume the mantle of system clock. The switchover is automatic and there's no dead time where nothing is happening.

When is this needed? As intimated earlier, PICs support a sleep command. When executed, the power hungry stuff is shut down and the chip idles in a very low current state. In particular, any crystal being used as a primary clock is halted. Various signals like a reset, an interrupt, or something called the watchdog timer can wake the chip up again. If you want some instant action while the crystal revs up, then consider the two-speed



wake-up just described. To enable this response, go to the IESO bit in CONFIG2.

The Experiments

With that, we've covered the essentials of clocking the PIC. There are other niceties you can worry about later, and that's what datasheets are for. At this point, you'll at least be able to handle most common situations, and be in much better shape to actually read that dang thing!

To really cement what you've learned in place, you're encouraged to try the baker's dozen of experiments described in the **sidebar**. You will literally learn by doing. **Figures 6** and **7** depict the hardware setup, while the PIC16F88 software is taken care of by the source code available from the download files at the article link. The programs are written in the free open source Great Cow Basic language, but are easily ported to PICBasic and other languages. In each experiment, the source code includes lots of comments to help you along, and also describes what to do in order to interpret the results.

Give these a try and see for yourself that clocking a PIC isn't nearly as gruesome as you might have once thought! **NV**

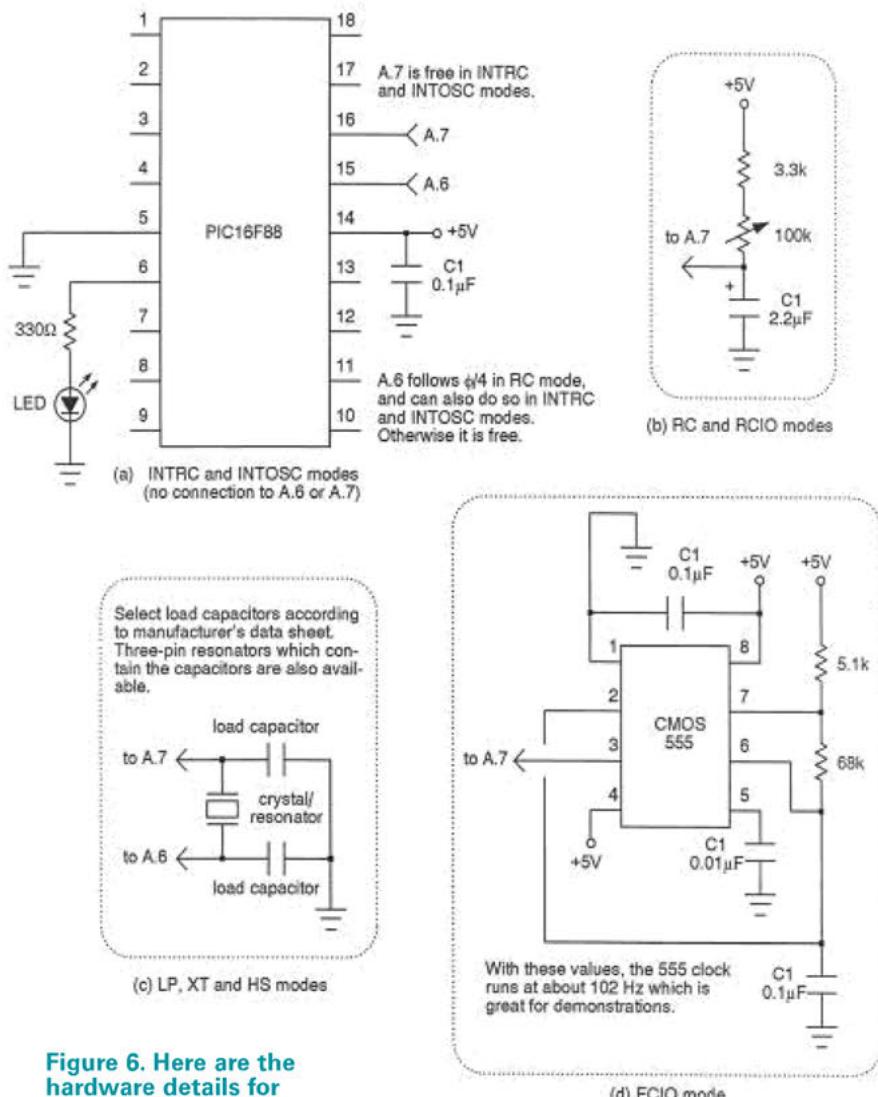


Figure 6. Here are the hardware details for primary clocks.



PBP3
PICBASIC PRO™ Compiler

Experimenter - \$49.95
Silver Edition - \$119.95
Gold Edition - \$269.95

The industry-standard BASIC compiler for Microchip PIC® microcontrollers.

Multi-Seat Licensing
for Educational Institutions
Upgrade from
PICBASIC™ Compiler (PBC)

Download a FREE trial version now.

www.PBP3.com

microEngineering Labs, Inc.

www.melabs.com

888-316-1753

PICBASIC and PICBASIC PRO are trademarks of Microchip Technology Inc. in the USA and other countries. PIC is a registered trademark of Microchip Technology Inc. in the USA and other countries.

#	Feature Illustrated	Source Code
1	INTRC, A.6 free	INTRC.GCB
2	INTRC, A.6 follows $\phi/4$	INTRC.GCB
3	INTOSC, A.6 free	INTOSC.GCB
4	INTOSC, A.6 follows $\phi/4$	INTOSC.GCB
5	RCIO, A.6 free	RC-RCIO.GCB
6	RC, A.6 follows $\phi/4$	RC-RCIO.GCB
7	LP, low power 32.768 kHz crystal	LP.GCB
8	XT, medium power 4 MHz crystal	XT.GCB
9	HS, high speed 19.6608 MHz crystal	HS.GCB
10	ECIO, 555 timer external clock	ECIO.GCB
11	Failsafe, yank crystal, and see INTOSC take over	FAILSAFE.GCB
12	Switching from secondary to primary on reset	SWITCHING.GCB
13	T1OSC as a secondary clock	T1OSC.GCB

List of Experiments.

The download file for this article contains the source code for 13 experiments using the common and inexpensive PIC16F88 microcontroller. The exercises demonstrate all of the various clock modes. The programs are well documented and contain instructions on how to set things up and what to look for. The hardware

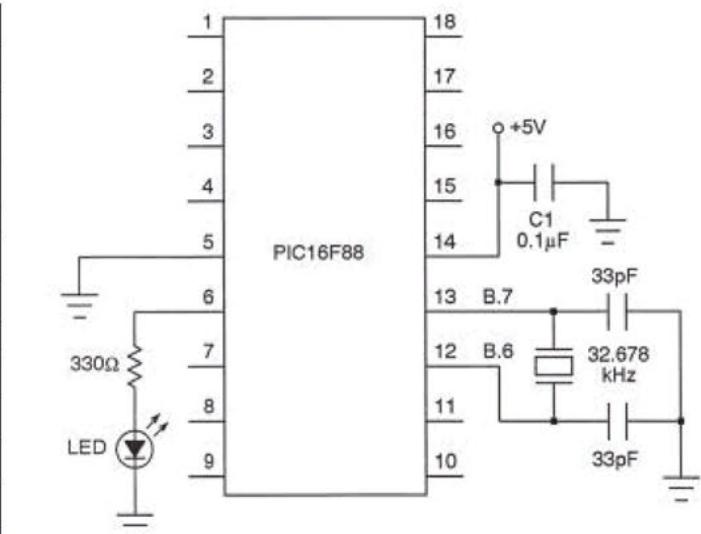


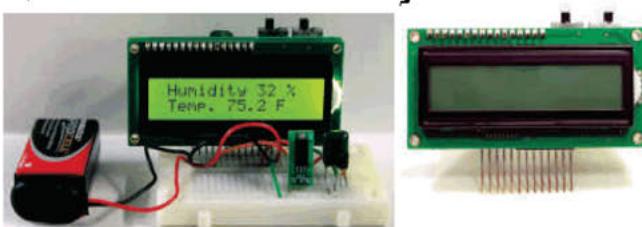
Figure 7. T1OSC mode uses a watch crystal as a timebase.

arrangement for Experiments 1 through 12 is depicted in Figure 6, while Figure 7 shows what's needed for Experiment 13.

In Experiments 2, 4, and 6, use an oscilloscope or frequency counter to monitor $\phi/4$. In all cases, an LED is blinked just to give the PIC something to do.

Images Scientific Instruments Inc.

PIC Basic Project Board



16F88 PIC Basic Project Board Features

- * LCD Display - back light & contrast control
- * 2 A/D channels, 8 digital I/O lines, 5V and 9V operation
- * Free student version of PICBASIC PRO & Microcode Studio with purchase of PIC Programming book bundle.
- * 16F88 Programming Service Available
- * Project library with source codes and hex files available.
- * See 16F88 PCB Project page for more details

Sensors:

- Humidity
- Temperature
- Toxic Gas
- Bend and Force Sensors
- CdS Light sensors

Do More:

- Frequency Meter
- Pulse Generator
- Elapse Timer
- Radiation Pulse Counter
- Volt Meter

www.imagesco.com/microcontroller/16f88.html

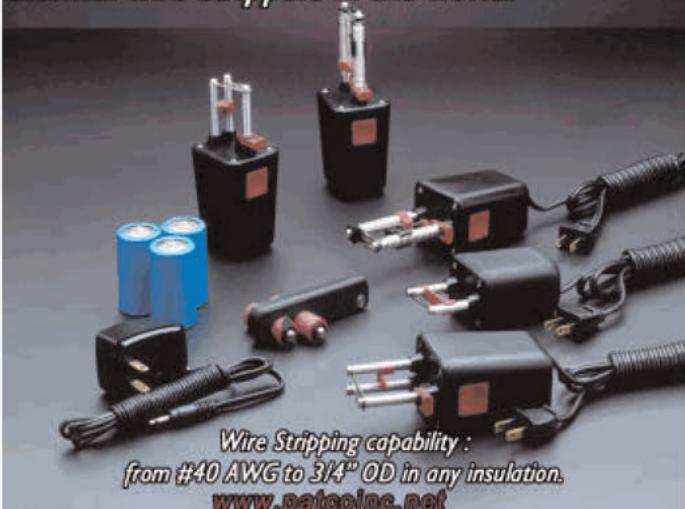
PATCO Inc.

Thermal & Mechanical Wire Strippers

PATCO, Inc.
5142 Rhodesdale-Vienna Rd.
Rhodesdale, MD 21659
410-444-4010
info@PATCOinc.net

Established 1979

Patco Inc. is the leader in hand-held wire stripping technology used in the electronic and electrical industry, and offers the largest variety of thermal wire strippers in the world!



Wire Stripping capability:
from #40 AWG to 3/4" OD in any insulation.
www.patcoinc.net

MakerPlot – Part 1

The DIY Software Kit

By John Gavlik and
Martin Hebel

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/january2014_MakerPlot.

Last month, we showed you how to connect an Arduino Uno to MakerPlot in order to plot a single channel of analog data using a 10K potentiometer to generate the variable voltage readings. We also showed you how MakerPlot can scale a microcontroller's raw 0 to 1023 A2D (analog-to-digital) output values into corresponding 0 to +5 volt levels – without resorting to coding the conversion math in the sketch. This time, we're going to show you how to add digital signals for plotting, as well as demonstrate how to play back both the analog and digital data in the plot area. Then, we'll show you how to data log what's being plotted to an Excel file for later analysis. If you haven't already done so, download the free 30 day trial copy of MakerPlot from www.makerplot.com to follow along. Let's get going.

Arduino Hardware Setup

Figure 1 illustrates the Arduino Uno with an attached 10K potentiometer. This hardware setup can plot a single channel of analog data. However, MakerPlot is capable of plotting up to 10 analog channels, so you're able to plot all kinds of analog inputs like audio signals, light sensors, thermisters, heart rate monitors, etc. – all at once. For this example, we'll keep it simple at one analog input for the pot.

Generating Digital Signals

MakerPlot is capable of plotting digital signals, too. **Figure 2** is a sketch for generating digital data along with the single channel of analog data from the potentiometer. While MakerPlot is capable of plotting up to 32 digital channels (bits), we'll plot just eight to show the concept. This is how it's done. Referring again to **Figure 2**, you can see that we added the variable "x" that's used in a *For Loop* with values between 0 and 255. We did this in order to produce the eight channels of digital data. Here's the part of the code that outputs the ASCII digital data. It's separated into

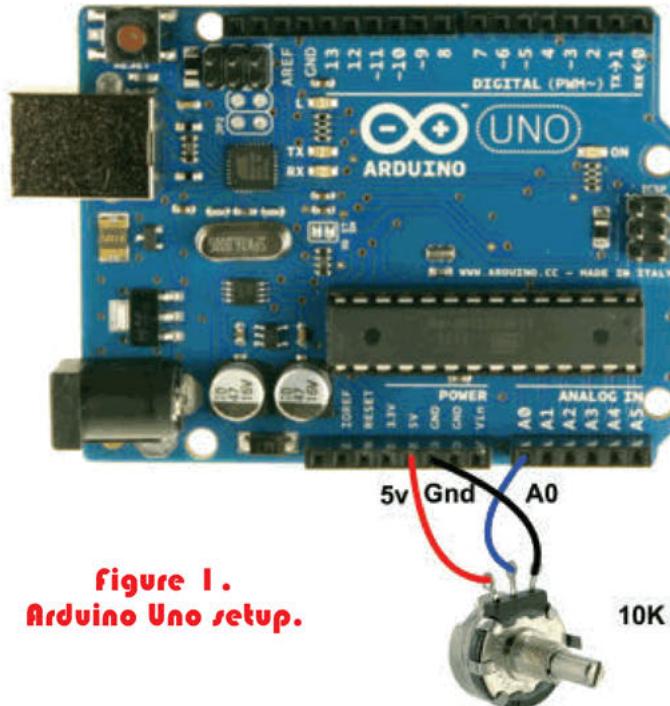


Figure 1.
Arduino Uno setup.

```

File Edit Sketch Tools Help
File Edit Sketch Tools Help
plotting_analog_and_digital_data
// Plotting Analog and Digital Data
// This sketch is in the public domain

const int analogInPin = A0; // Analog input pin that the potentiometer is attached to
int sensorValue = 0; // value read from the pot
int x; // value for digital data

void setup() {
  // configure hardware
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}

void loop() {
  for (x = 0; x < 255; x++) {
    // read the analog in value
    sensorValue = analogRead(analogInPin);

    // print the analog values formatted for MakerPlot
    Serial.println(sensorValue); // send analog value with CR

    // print the digital values formatted for MakerPlot
    Serial.print("%["); // configure for digital data with %[ 
    Serial.print(x); // current digital data value
    Serial.println(" ADC 8]"); // use ADC 8 instruction to make 8 digital channels
    // end with closing ] and CR

    // wait 100 milliseconds
    delay(100);
  }
}

```

Figure 2.
Arduino
sketch.

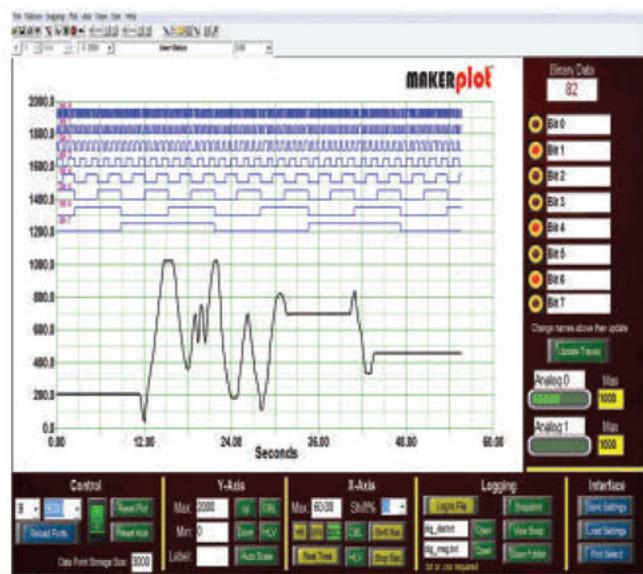


Figure 3. Digital interface plotting analog and digital data.

three lines for a purpose. Notice how the entire string is bounded by double quotes (" "); this is important so that the resulting data within the quotes are translated into ASCII:

```

Serial.print("%[");
  // configure for digital data with %[ 
Serial.print(x);
  // current digital data value
Serial.println(" ADC 8]");
  // use ADC 8 instruction to make 8
  // digital channels and end with
  // closing] and CR

```

In the first line, MakerPlot uses the percent (%) sign to distinguish what follows as digital data. Next, comes a square open parenthesis ([) that MakerPlot uses to bound the variable. The next line is the value "x" that produces the digital output. That, in turn, is followed by the MakerPlot instruction (**ADC 8**) that tells MakerPlot to generate eight channels of digital data from the x value. Then, we round out the data set with a square closed parenthesis (]). The **Serial.println** instruction also terminates the analog and digital strings with a Carriage Return character which MakerPlot requires to know when the data for this field is complete.

Now that the sketch is complete, we'll copy and paste it into the Arduino IDE (Integrated Development Environment), verify it, then load it into the Uno. Then, we're ready to do three other things:

1. Plot the analog and digital data.
2. Replay it in the plot area.
3. Data log it.

Plotting Analog and Digital Data

For this sketch, we're going to choose the **MakerPlot Digital Interface** (Figure 3) since it plots both analog and digital signals, with the emphasis on digital plotting. With the Uno physically connected to the PC via the USB cable, we've clicked on the rocker switch at the bottom-left (under **Control**) to begin plotting the data. As you can see, the digital data are at the top (LSB first) and the analog (potentiometer) data are on the bottom. The pot shaft has been adjusted back and forth to generate a data plot.

The vertical Y scale is between 0 and 2000 to accommodate the raw A2D data going from 0 to 1023. The Y scale can be changed to any value you choose using the menu controls on the bottom. You can also witness the analog data as **Analog 0** in the bar graph on the right side of the plot (Figure 4). The horizontal bar graph range is currently set to 1000, meaning each one of

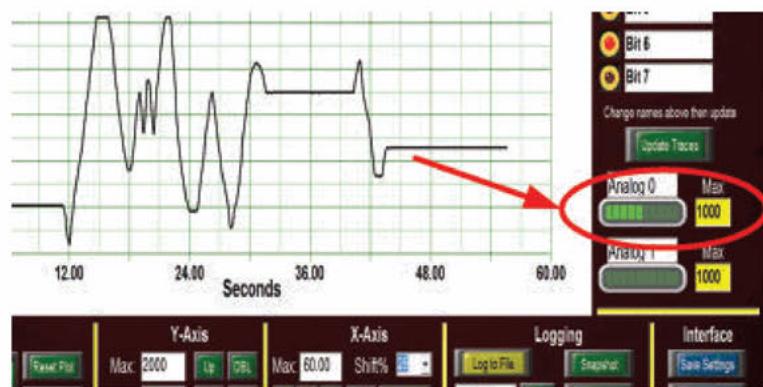


Figure 4. Analog 0 bar graph.

the 10 LED icons that make up the bar graph is worth 100 analog units. Like the Y scale, you can change this value to anything you wish — including changing the title from “**Analog 0**” to, say, “**Pot Value**.” Just key in the new range (1023) and title (**Pot Value**), and that’s it! **Figure 5** shows the results of this customization effort.

The vertical Y scale has no real meaning for the digital data since each channel or bit of digital data simply plots

along successive horizontal grid lines — in sequence — from LSB (top) to MSB (bottom). Also notice that each digital trace is labeled as bit 0 through bit 7 — both on the left side of the plot area and, also, right next to the LED that corresponds to the digital data channel on the right (**Figure 6**).

Like most everything in MakerPlot, these designations can be changed to whatever you want; you can name each digital channel to represent its meaning within your data monitoring setup. For example, the first and last channels to represent the LSB and MSB of the bits have been relabeled (**Figure 7**). Just key the new names into the text boxes next to the LEDs, then click the **Update Traces** button below. It’s that simple!

Playing Back Data

Now that we have analog and digital data plotting, let’s show you how to replay the last 3,000 data points so you can better analyze it. The playback “capture” is based on the **Data Point Storage Size** indicated in the **Control** menu area (**Figure 8**). Of course, you can change this value up or down to suit your recording needs; the larger the number, the greater the number of data points captured. However, the larger the number, the longer it takes for MakerPlot to refresh the plot area. Practical **Data Point Storage Size** values range from 500 to 50,000 with 3,000 being the default. So, for this example, we’ll leave it at the 3,000 default setting.

The captured data — which is based on the **Data Point Storage Size** — is cleared when the plot resets as it gets to the right-end of the plot area. So, to execute a recording, first let the plot run from left to right and “nearly” to the full length of the plot area — but not all the way. Just before it gets to the end, click the green rocker switch (**Control** menu) to disable the connection between the Uno and MakerPlot. For this example, the rocker switch was clicked to red (OFF) about 50 seconds into the plot, and just before the plot reached the right-hand side where it gets reset. Next, the **Player** icon in the **Toolbar** was clicked (the one that looks like a “hand and wand”), which brought up the **Player** control (**Figure 9**). So, now with a clear plot area and the **Player** control displayed, here’s how to replay the last set of captured analog and digital data.

Click on the **Play** button inside the **Player** control. This will start the data playback (**Figure 10**). The **Play** button changes to **Pause**,

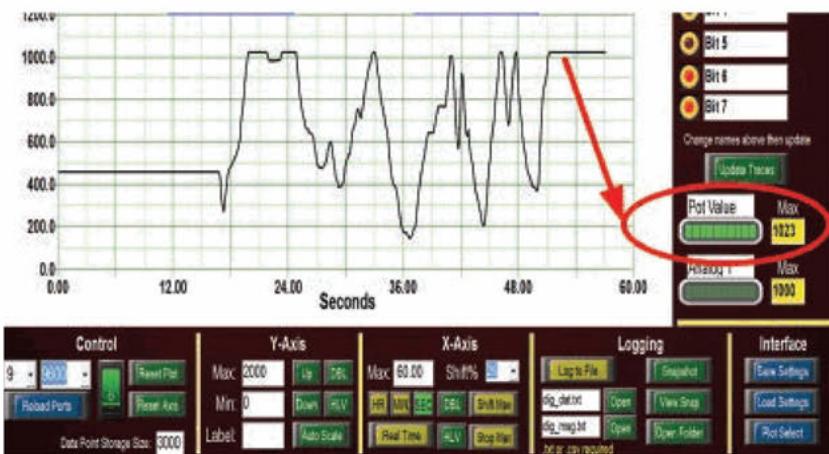


Figure 5. Adjust bar graph and range value.



Figure 6. Digital channel text annotation.

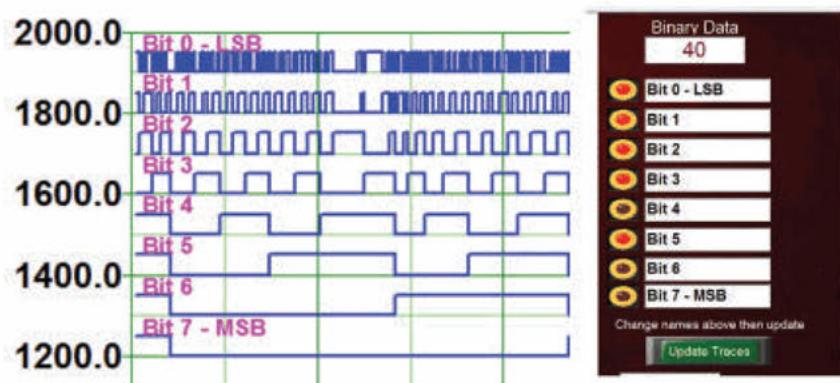


Figure 7. Revised annotation.

meaning that you can click it again to halt the playback in order to examine a particular area or point in the plot. Clicking it again resumes the playback. You can see the progress of the playback by looking at the **Data Position** slider arrow – AND – you can place your mouse cursor on the **Slider** arrow to move it back and forth which allows the plot to replay up to the point where the slide arrow comes to rest. Plus, you can click the **Loop** check box for continuous play.

To slow things down, you can place your mouse cursor on the **Delay** arrow and move it to the right which adds the indicated millisecond delay between plotted data points. Try it. With enough delay, it will give you a “burst by burst” plot of the recorded data so you can stop it wherever you like with much better control.

So, that's how to use the **Player** function to replay data for immediate review. Now to data logging in Excel CSV format.



Figure 11. Logging menu.

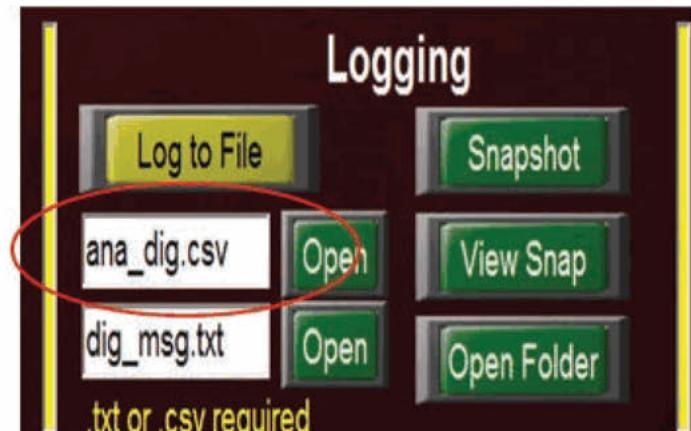


Figure 12. Logging menu with new file name.



Figure 8. Data Point Storage Size.

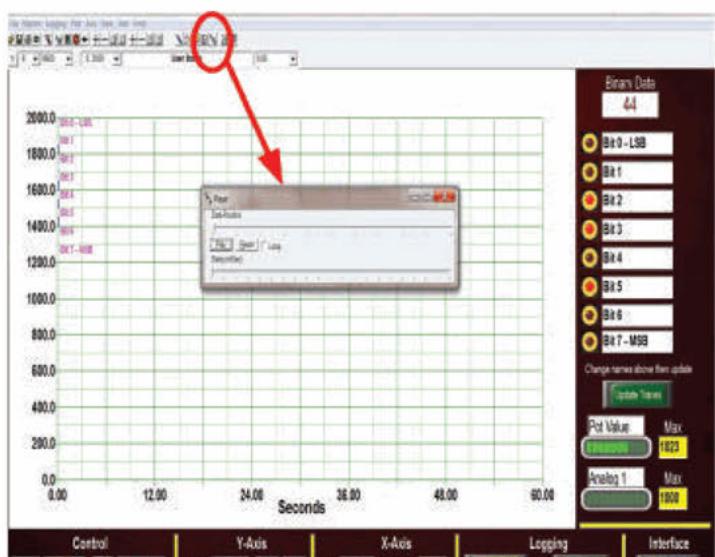


Figure 9. Bring up Player tab.

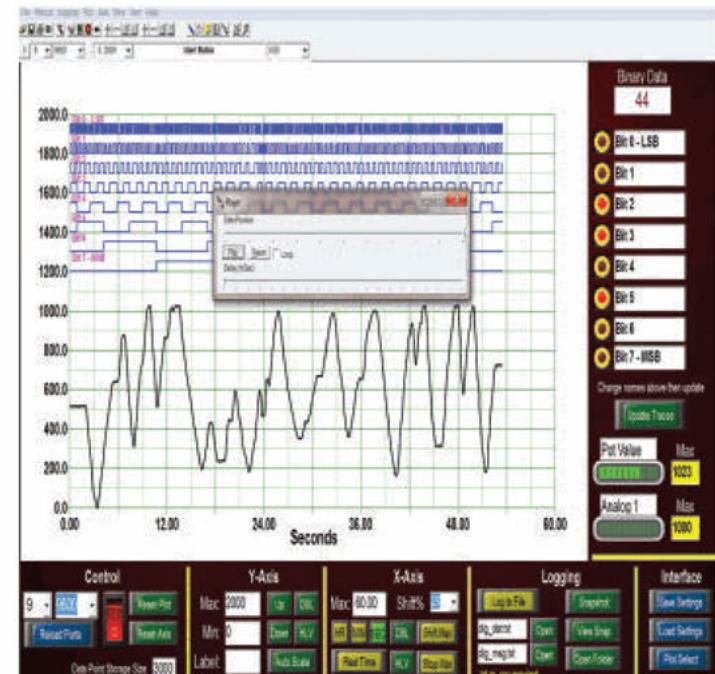


Figure 10. Data replay.

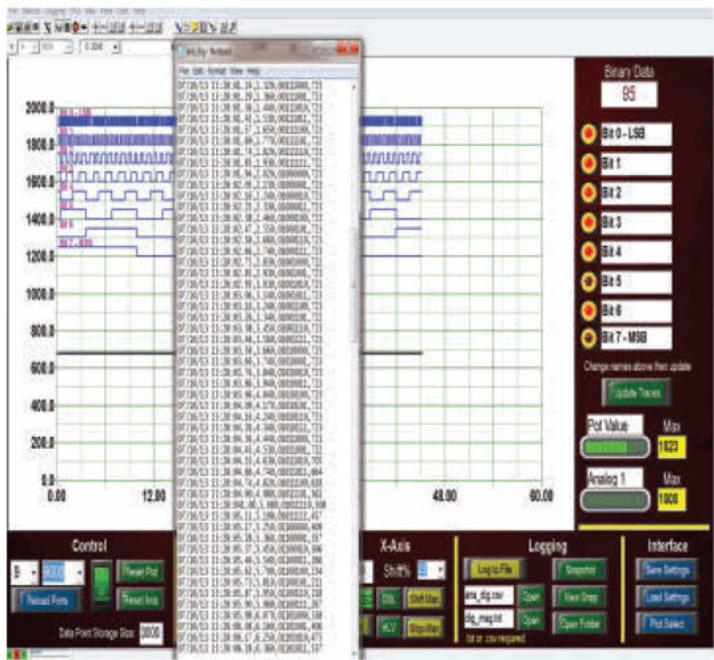


Figure 13. Logged data in NotePad.

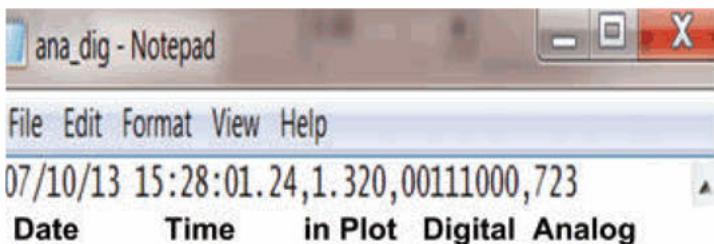


Figure 14. Typical data log record.

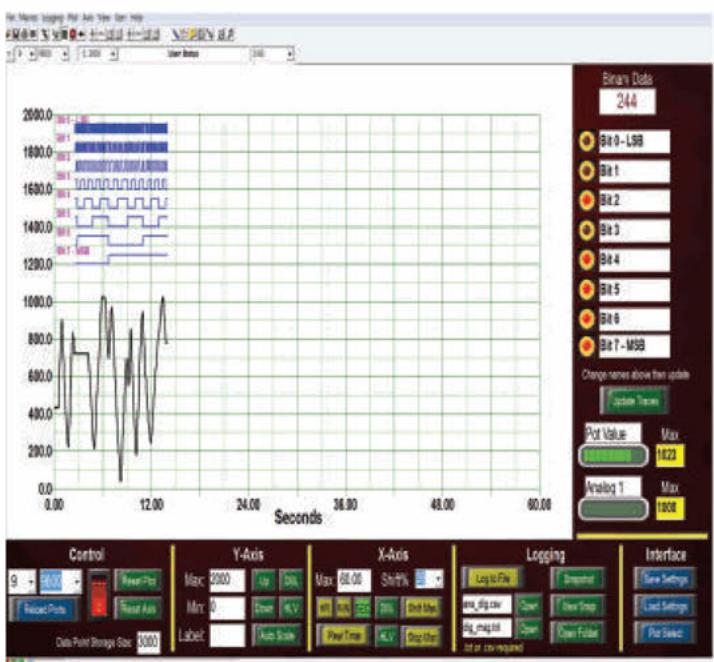


Figure 15. Plot of logged data.

Data logging

With data logging in MakerPlot, you're not limited to the last 3,000 or so data points; you can log any amount of analog and digital data that you choose for as long as you choose — minutes, hours, days, etc. — to either a text file or a CSV file. Remember, you're basically saving a text file to a computer hard disk and not just a few memory chips on your micro's board. So, there are no practical memory-limiting issues to worry about when data logging with MakerPlot. Here's how it's done.

First, take a look at the **Logging** menu at the bottom in **Figure 11**. On the left side, you'll see the **Log to File** button and below it, two text boxes with file names. The first text box is for logging data and the bottom text box is for logging messages. Messages, by the way, are an important alerting feature since you can program MakerPlot to output messages based on certain data levels or combinations, and then you can log these messages with date and time stamps for later analysis. We won't be getting into logging messages in this example; however, just be aware that you can.

The default data text file is **dig_dat.txt** which means that logged data will be sent to this file name which — by the way — is a standard text file with a .txt extension. We'll change the file name and extension to reflect logging our analog and digital data to the CSV file. Once again, this is easily done by simply clicking on the text box just below the **Log to File** button and keying in a new file name. The one chosen is called **ana_dig.csv** (**Figure 12**). With that done, let's log some data.

Click the red rocker switch in the **Control** menu to start the data flowing into MakerPlot. Now, click on the **Log to File** button which will change from yellow to green. As long as it stays green, every data packet from the Uno will be logged and time stamped. Let a few seconds go by and then click the **Log to File** button again; this ends the data logging session. Now, let's look at what got logged.

Click on the **Open** button next to the **ana_dig.csv** text box. This brings up the NotePad file of the logged data (**Figure 13**). Here, we can examine the logged data in some detail. Looking at the first record (**Figure 14**), we see that the numbers on the far left represent the date and time (with time down to hundredths of a second). This is followed by the time into plot (in thousandths of seconds) followed, in turn, by the state of the eight digital data bits (MSB to LSB), then is followed by the analog value of the potentiometer. If there were more analog channels for logging, they would follow this one. As you can further see, these individual fields are separated by commas which will allow for direct transfer to Excel.

Where, you might ask, is this file on the hard disk? When you install MakerPlot, the default location is **My Documents → MakerPlot → Data** which is where you'll find it. Just make sure to turn on your file extensions to

see the .csv extension. You can now load the **ana_dig.csv** file into Excel for any type of analysis.

We're not done with what MakerPlot can do with this file just yet, so let's show you a few more interesting things.

Displaying logged Data

To display the logged data, let's first turn off the data flowing from the Uno by clicking the green rocker switch in the **Control** menu. Next, click on **Logging** on the top line, followed by **Plot From Data Log**. Then, click **Open** (**Figure 15**). What you'll see is the logged data plotted just as it was recorded.

Playing Back logged Data

Going back to the **Player** function, you can replay the logged data just as before, but this time we're replaying logged (not real time) data (**Figure 16**). This gives you an opportunity to examine the logged data in graphic detail. You can also expand or contract the time axis using the **DBL** and **HLV** buttons to your liking. You can shift the plot left and right with the arrow keys on the Tool Bar, as well.

There are videos on the MakerPlot website that show you all about the **Player** function and data logging features. Just go to the MakerPlot website (www.makerplot.com) and click on **Basic Plotting → Player Tab** and **Basic Plotting → Logging Menu** to view all that can be done (**Figure 17**). These short videos will give you a much better idea of these features than we can do here in just a few words and figures, so be sure to check them out.

Conclusion

That's about all the space we have for this time, so let's review what we've covered and what's coming up in the next article.

We learned how to plot both analog and digital data from the Uno, followed by playing it back using the **Player** function. We also learned how to log the data and play it back on the plot area for greater analysis. Finally, we showed you how to configure the data logged file name with a .csv extension so it can be ported directly into Excel.

In the next article, we'll continue to expand on our Uno setup to show you how you can add two pushbuttons and an LED to your Uno board in order to begin doing some interesting bi-directional control

experiments. This will take more than just one article to address as it covers a lot of ground; however, it will be worth it since you'll begin to understand how MakerPlot can act like a front panel for your microcontroller projects with switches and other controls that can manipulate your micro's actions. **NV**

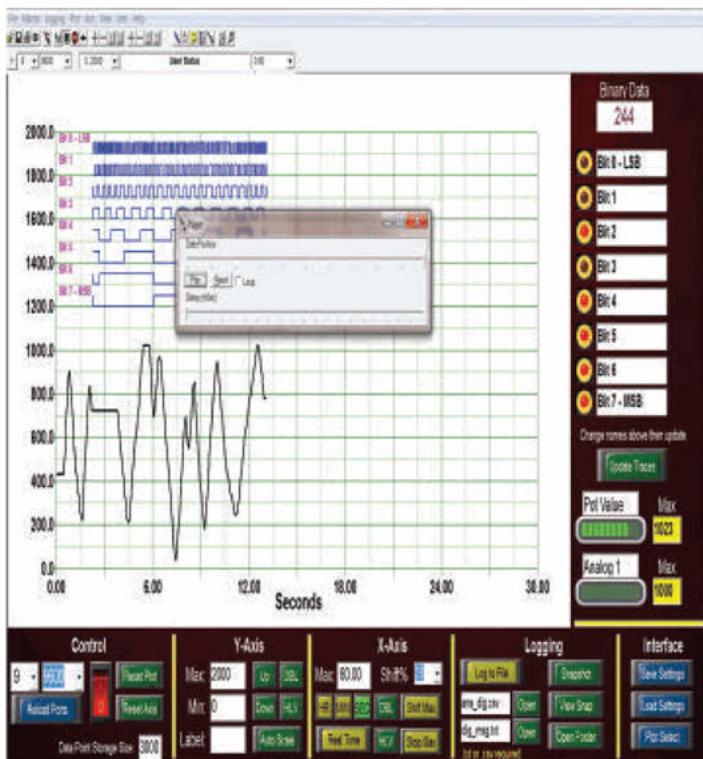


Figure 16. Playback of logged data.

The figure shows two windows of the MakerPlot software. The left window is titled "Player" and contains a "Data Position" section with "Play", "Reset", and "Loop" buttons, along with a "Delay (mSec)" slider. The right window is titled "Player Tab" and contains a descriptive text block: "The Player Tab allows you to record and play back plotted data. This video will show you how to setup MakerPlot to do this as well as how to save your recordings for later playback and analysis." Below these are two more windows: the "Logging Menu" which lists various logging options like "Save Data to File" and "Delete Data File", and a "Help" window with the text: "Data logging is an important MakerPlot feature and the Logging Menu is where you can configure how to log data and where to store the logged files. Data logging can be by time or by data points with several options that make it a versatile logging tool."

Figure 17. MakerPlot video tutorials.

READER FEEDBACK

Continued from page 7

bias resistors. On the testing unit, two pairs of precisely matched millimeters exhibited each tube's plate and screen element. Another single meter (a VTVM) was employed to read control grid emission/gas with a toggle switch to select either tube. There was also a DC voltmeter to read the B+ power supply which was

variable in order to match different tube types. There was a standby switch in B+ only. The unit housed a 5" speaker to indicate noise. Also, the unit had two preheat sockets (elements only) which hastened the interchange of tubes.

Over the years, I added nine-pin tube socket adapters. We could

match 807, 5881, 6L6G, 6V6GT, 6CA7, 6550, 6BQ5, etc., tubes. During the matching process, the tubes were run hard with high plate current (thanks to the various cathode resistors and adjustable B+) to achieve class "A" test levels.

Many of my Soundmaster amplifiers are still around today and have become high priced collector's items. With over 30,000 built and sold under warranty, I expected that a certain percentage would be returned to the factory for repair. That work — which I thought might keep me busy in later years — never materialized. I have, however, been able to see one of my amps in use after 60 years and, remarkably, the output tubes still read as balanced with satisfactory frequency responses and noise levels.

In addition to highest quality parts and excellent hand-wiring, the careful balancing of the output tubes with my tube matcher was definitely an important reason for our success. It served me well over the years. We tested so many tubes with it that I had to replace sockets on occasion. According to my calculations, we matched over 100,000 tubes.

Hy Bloom, CEO (Retired)
Soundmaster, Ltd.
Ottawa, Canada

Christmas Can Concerns

(The following feedback is in regards to the Christmas can light dimmer article in the December 2013 issue.)

Although I like the article's concept, there is a safety issue here. Even though the neutral wire is connected to earth ground at the circuit breaker box, devices should never rely on the neutral wire for a safety ground as was done in this project. Since the dimmer enclosure is aluminum, a shock hazard exists if the 120 volt outlet hot and neutral wires are reversed (which is not uncommon) because the aluminum enclosure would then be at a 120 volt potential. Nowhere in the article does the writer verify that the outlet is wired properly. Using three-

proBOTIX CNC Routers & Control Systems

Imagine What You Can Make or... Make What You Can Imagine!

Introducing the PBX-BB BeagleBone Powered CNC Controller!

www.probotix.com

Repair Monitors, Power Supplies, Gameboards? Check any Electrolytic Capacitor In-Circuit with 100% Accuracy in 3 Seconds---GUARANTEED*

Automatically discharges capacitor

Checks DCR with alerts for shorts

Measures DCR to 500 ohms

Measures ESR from 0.1 to 20 ohms

Checks caps from .47uF to 2200uF

Beeps one to five beeps for quality

Three-color chart for good-fair-bad

*range 0.47uF - 2.2KuF 90-day money-back guarantee

Portable, Easy to Use, Inexpensive, Highly Recommended by leaders in the industry. **CapAnalyzer 88A** by EDS, Inc.
At your favorite distributor, or call 561-487-6103 www.eds-inc.com

conductor connectors would prevent the potential hazard. Of course, there would be no hazard if the dimmer enclosure was plastic or some other non-conducting material.

Ed Terry

Please be advised that what this article suggests is a hazard of electrical shock. It suggests using a two-wire zip cord for this project and even though it indicates the use of a "polarized two-pole plug" it implies that it is a safe thing to use. The article indicates that the wider prong on the plug is the "ground" connection.

This is far from being true! It is a current carrying conductor called neutral — the return path back to the electrical panel. It may be tied to ground at that point, however, the farther away from that connection the device is, the wires are adding resistance to the circuit. The more current being drawn through this resistance then causes a voltage to appear at the connection (even though it may be small); it is enough to create a hazard to anyone or anything touching the can. When the lights are on, the current supplying the device needs to return to the panel it originated from via that neutral wire which is now connected to the can.

In the case of GFI circuits, this connection would cause it to trip anytime someone touched it and was actually grounded themselves because of this connection.

In the case of someone in a refurbished house (by other unknown persons), they may have the wires crossed when they installed a new plug, etc.

The real problem here is the metal can being used, and being tied to the neutral wire as a safety ground is NOT safe. It would have been much better to have used a plastic box or plastic can and not ground it at all. (Plastic PVC gray boxes can be purchased at Lowes or Home Depot at a very small price.)

The better choice would have been to use a three-wire cord and actually use the green ground wire to ground the can for safety.

Not everyone reading this article is an electrician and may not connect the correct wire to the can since zip wire is not color-coded.

As it shows in the simple schematic drawing on page 44, there's a plain two-wire circuit indicating one wire as hot and the other as ground. This is not so simple as seen on the paper drawing. Please advise your readers of this and the author/provider of the article.

David W. Bent

All excellent points. The safest approach is to either use a plastic fully insulated enclosure that prevents user contact with either hot or neutral, or to use a metal enclosure connected to true ground through a three-prong power cord. As pointed out, it's possible that even if someone used a polarized plug the house wiring could be reversed, resulting in connecting the hot lead to the metal

Continued on page 81

Feedback Motion Control

The Old Way

- 1) Build robot
- 2) Guess PID coefficients
- 3) Test
- 3a) Express disappointment
- 3b) Search Internet, modify PID values
- 3c) Read book, modify PID coefficients again
- 3d) Decide performance is good enough
- 3e) Realize it isn't
- 3f) See if anyone just sells a giant servo
- 3g) Express disappointment
- 3h) Re-guess PID coefficients
- 3i) Switch processor
- 3j) Dust off old Differential Equations book
- 3k) Remember why the book was so dusty
- 3l) Calculate new, wildly different PID coefficients
- 3m) Invent new, wildly different swear words
- 3n) Research fuzzy logic
- 3o) Now it is certainly not working in uncertain ways
- 3p) Pull hair
- 3q) Switch controller
- 3r) Re-guess PID coefficients
- 3s) Switch programming language
- 3t) Start a new project that doesn't need feedback control
- 3u) See parts in box. Feel guilty. Go back to old project
- 3v) Start testing every possible combination of PID coefficients
- 3w) Apply eye drops to red, blurry, sleep-deprived eyes
- 3x) Wait, it's working!
- 3y) Decide not to do any more projects that require control options
- 3z) Wonder why someone doesn't just make a thing that tunes itself

The Kangaroo x2 Way

- 1) Build robot
- 2) Press Autotune
- 3) Get a snack

Kangaroo x2
adds self-tuning
feedback to SyRen
and Sabertooth motor
drivers.



\$24.99

DimensionEngineering
www.dimensionengineering.com/kangaroo

Hamboree Mini Maker Faire

Saturday, February 1, 9AM - 5PM

HamboreeMakerFaire.org

Miami Dade Fair Expo Center - 10909 SW 24st Miami, FL 33165

**** Double Feature ****



Presented by:
DADE RADIO CLUB
OF MIAMI INC.

2014 TROPICAL
HAMBOREE
JAN 31, FEB 1 - Miami, FL

<http://hamboree.org>

The Biggest Ham Radio, Electronics and
Emergency Communications
Show for South Florida, the Americas
and the Caribbean

The Arduino Classroom: Arduino 101

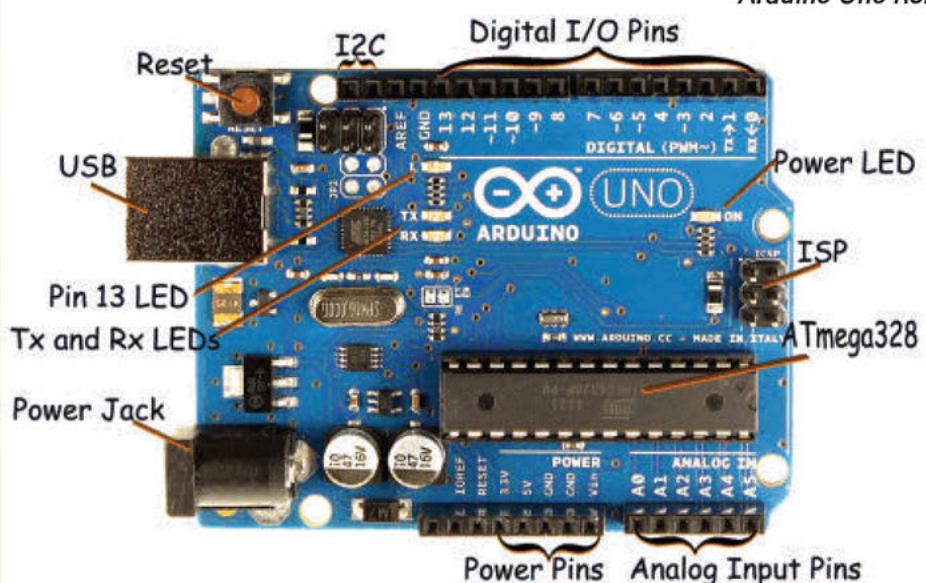
For a few years now, this column has wandered around the world of AVR microcontrollers guiding folks in various aspects of C programming and designing embedded systems. I have received lots of great feedback and one thing that seems apparent is that many want a more formal approach to learning this stuff. Getting everything in bits and pieces just isn't cutting it for some. The Internet is overflowing with informal cookbook projects and tutorials to solve every little problem, but what seems lacking are stepwise approaches like one might get in a regular classroom. If you are making LEDs blink just fine but have a nagging doubt whether you really know what you're doing, then maybe it's time to follow a curriculum that can take you through each step so that when you finish, you'll know you've covered all the bases.

Changing Directions

That is exactly what is going to happen over the next who-knows-how-many articles: a formal curriculum starting with Arduino 101 where you learn computing and electronics basics much like you'd learn if you took a standard semester-based introductory course. After that, we'll take this base of knowledge and use it as a platform to move up to Arduino 102 where we will learn how to turn an idea into a product. From there, we'll move on to Arduino 103 where we'll learn to design and build a data logger and a robot, thus completing a solid introduction to the fundamental concepts of computing and electronics as they apply to microcontrollers and embedded systems.

To help in this, I've started www.arduinoclassroom.com where each of these magazine articles will be presented along with laboratories, exercises, quizzes, and a forum.

FIGURE 1: The Arduino Uno R3.



Now instead of being a passive reader, you get a chance to have some interaction and to talk back.

But First ... HELP!

Sure. I'm going to create a magazine/Internet based classroom that will bridge the gap between real classes and all the stuff the Internet has to offer and I'm going to do it all by my lonesome. Not! I'm not that talented (or crazy). This is only going to work if I get help from students and teachers. I'll need some novices to decide to be dedicated students and tell me where they are having difficulties with this stuff. Also, I'll definitely need some teachers to wade through this and offer suggestions as how this can be done better to meet what they know are the real needs.

So far, I've got my most excellent friend Jay Flanders helping out. Jay taught electronics at Kaskaskia College in

Centralia, IL, so he has a lot of great insight to provide (plus, he laughs at my stupid jokes which helps keep me moving forward). If you are a teacher and want to discuss these articles and/or the website, please contact me at teacher@arduinoclassroom.com.

The version of articles on the website will have pretty much the same text fancied up a bit with accordion folds that let you open each section separately. It will also have check boxes that let you mark each text section, exercise, lab, and quiz as you complete them so you can keep track of where you were at the end of each study session if you prefer to do that online.

Let's get to the first article ...

Chapter 1: Getting Started

Why do you need the Arduino Classroom?

The official Arduino website is located at www.arduino.cc (yes, that is .cc and not a typo). With that website and reasonable Google skills, you can find everything you could possibly need or want to know about the Arduino — so why do you need this Arduino Classroom?

Well, if you've spent much time wading through the material available on the Internet, you've seen that it is of varying quality and not well organized to promote stepwise learning. What you want is a logical progression for the presentation of information — a great reason to use a series of articles like this. Together, they organize the materials for a sensible learning sequence and present labs with tested examples using a hardware projects kit.

Having this structure helps you move quickly through what you need to know, and provides a base of knowledge and skills that you can use to pursue more advanced projects using what you find on the Internet. So, let's get started!

What is a microcontroller?

A microcontroller is a very small computer that has digital electronic devices (peripherals) built into it that help it control things. These peripherals allow it to sense the world around it and drive the actions of external devices. An example of a use for a microcontroller is to sense a temperature and — depending on the value sensed — it could either turn on a fan if things were too warm or turn on a heater if things were too cool.

You might already be aware that microcontrollers are

in common devices like cell phones, microwave ovens, and alarm clocks that have buttons for you to input information and displays to tell you things. There are even more microcontrollers embedded in things where you never see them.

For example, there are 30 or more microcontrollers in an automobile. These do everything from sensing the oxygen intake to setting the fuel air mixture to measuring the cabin temperature for controlling the air conditioning levels.

What is an Arduino?

Massimo Banzi, the Arduino team leader, begins his book, *Getting Started with Arduino*: "A few years ago, I was given the very interesting challenge: teach designers the bare minimum in electronics so that they could build interactive prototypes of the objects they were designing." He summarizes his philosophy of 'learning by tinkering' with a quote from www.exploratorium.edu/tinkering: "Tinkering is what happens when you try something you don't quite know how to do, guided by whim, imagination, and curiosity. When you tinker, there are no instructions — but there are also no failures, no right or wrong ways of doing things. It's about figuring out how things work and reworking them."

Arduino provides a great toolset for designers, tinkerers, and anyone who sometimes just wants to play with an idea that uses electronics. The genius of Arduino is that it provides just enough access to get specific tasks done without exposing the daunting underlying complexities.

What can you do with a microcontroller?

An Arduino looks like the board shown in **Figure 1**. There are lots of varieties of Arduinos that get embedded into many interesting applications. Let's look at a few of these.

UAV

Chris Anderson, former editor of *Wired* magazine, developed an Arduino derivative for remote control aircraft — the ArduPilot for DIYDrones.com (see **Figure 2**). These are fully autonomous Unmanned Aerial Vehicles



■ FIGURE 2: Arduino-based UAV.



FIGURE 3: Turn signal jacket for bikers.

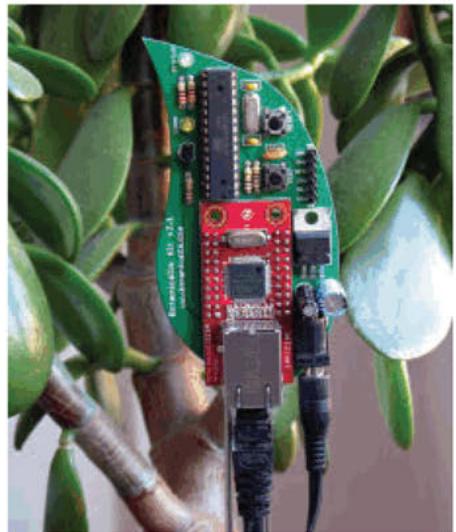


FIGURE 5: Botanicals.

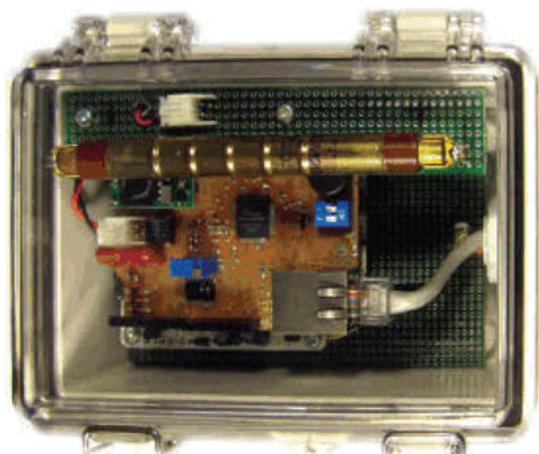


FIGURE 6: Internet connected radiation detector.

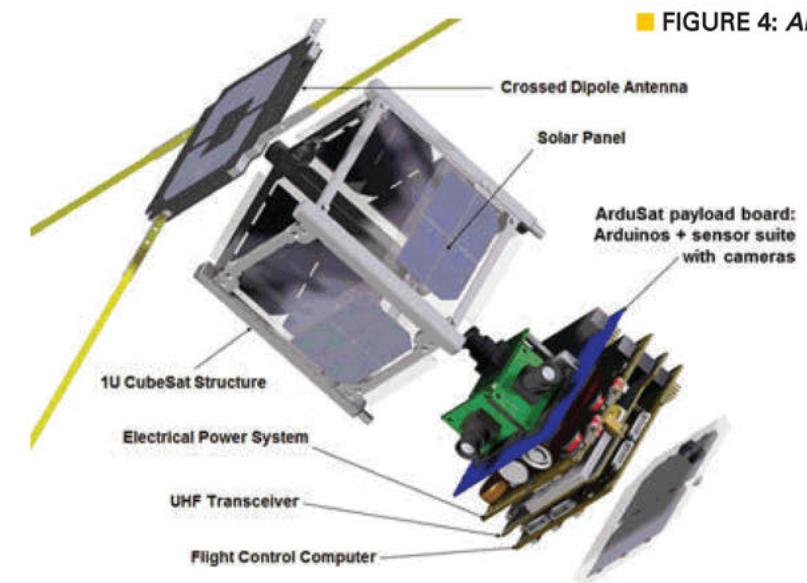


FIGURE 4: ArduSat.



Below the average public space geiger reading for Japan (0.081 μSv per hr).

Above the average public space geiger reading for Japan (0.081 μSv per hr).

10x above the average public space geiger reading for Japan.

100x above the average public space geiger reading for Japan.

FIGURE 7: Internet radiation reports.

(UAVs) capable of performing a mission without a pilot in control.

Electric Clothing

Leah Buechley, an associate professor at the MIT Media Lab, developed a variety of the Arduino called the Lilypad (shown in **Figure 3**) that can be used with fabrics. One clever application was to put it on the back of a jacket to provide turn signals for bikers.

Arduinos in Space

Why not? Take a look at **Figure 4**. Arduinos can be incorporated into near space missions.

Tweet Your Plant

Botanicals is an Arduino clone (shown in **Figure 5**) that lets a plant send you a Twitter message when it needs water.

Radiation Detectors on the Internet

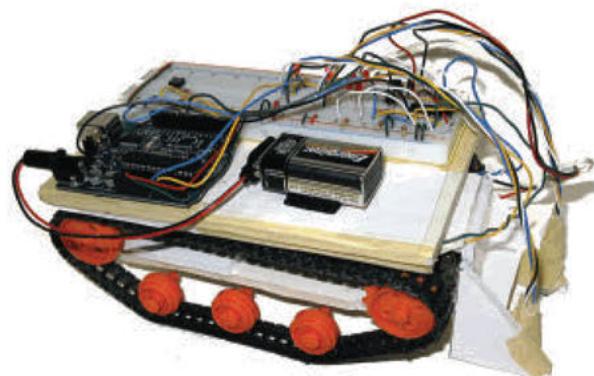
While getting a twitter message from your philodendron telling you it's thirsty is pretty cool, how about having a bunch of concerned citizens who aren't confident that the government is telling them the truth about radiation exposure getting



■ FIGURE 8: *Pebble prototype.*



■ FIGURE 9: *Pebble product.*



■ FIGURE 10: *Experimental robot.*

together and setting up an Internet-based monitoring system themselves? Well, that is exactly what happened after the tsunami at Fukushima, Japan that caused the local nuclear power plant to fail. The folks at www.tokyohackerspace.org designed a Geiger counter (**Figure 6**) based on the Arduino and figured out how to get it to report radiation levels on the Internet (**Figure 7**).

Pebble: Really Smart Wristwatch Project

Pebble is a wristwatch that uses an e-paper display and a wireless connection to coordinate its activities with either an Android phone or iPhone. This device does a lot more than just tell time; it can control your music, display email alerts, monitor time and distance when you run, and lots of other things you can see at www.getpebble.com. An Arduino and cell phone parts were used to build the prototype shown in **Figure 8**.

The designers started a Kickstarter project to get funding to go into production and though they requested \$100,000, they got a record breaking \$10,266,846.

Figure 9 shows the production Pebble.

Robots

Lots of robots are made with the Arduino; just Google ‘Arduino robot’ and see for yourself. **Figure 10** shows an inexpensive tracked robot mainly made of foamcore board, hot glue, and tape. It uses an Arduino and motor drivers on a breadboard and line/edge sensors taped to the front. It is simple, cheap, and quite effective for demonstrating basic robotic principles.

These are just a few of the many things you can do with a microcontroller such as the Arduino. Spend some time on the Internet researching the Arduino and you will see that the original question, “What can you do with a microcontroller?” might be better stated as, “What can’t you do with a microcontroller?”

Lab 1: Visit the Arduino Website

Required tools:

A computer with access to the Internet.

Estimated time for this lab: 30 minutes

Check off when complete:

- The Arduino website is divided into six main sections. First, visit the main site arduino.cc/en/Main.
- Visit the Getting Started section and get acquainted with what they have to say about themselves.
- Visit the Learning section and note the extensive list of links to projects. This will come in handy later when you want to do things not discussed in this curriculum.

- Visit the Reference section. This provides links to the documentation for the Arduino functions library. We will learn about a lot of what is shown here.
- Visit the Blog at arduino.cc/blog. This is where the team members post interesting things about the Arduino.
- Visit the Playground. This is a wiki where Arduino users can contribute. This is another huge resource that you will want to use later as you gain more experience with the Arduino.
- Visit the forum. This is a great place to get good answers to your questions. However, if you aren’t familiar with typical forum etiquette, I strongly recommend you first visit ‘How to ask questions that have a better chance of getting an answer’ at www.catb.org/~esr/faqs/smarter-questions.html.

Lab 2: Download the Arduino Software

You can get the Arduino software free from the Arduino website.

Required tools:

A computer with access to the Internet.

Estimated time for this lab: 30 minutes

Check off when complete:

Go to arduino.cc/en/Main/

Software and download the latest software for your particular computer system. The Arduino software is often upgraded, so you may see some differences in what is shown here and what actually happens, but the Arduino folks have been very good at keeping things simple and backwards-compatible so the differences shouldn't be major.

The following is for a Windows system as shown in **Figure 11**. The download for the Mac and Linux will be a bit different.

Click on your operating system.

Save the file. **Figure 12** shows the Windows dialog box

Download

Arduino 1.0.1 (release notes)

- + [Windows](#)
- + [Mac OS X](#)
- + [Linux: 32 bit, 64 bit](#)
- + [source](#)

FIGURE 11: Select your operating system.

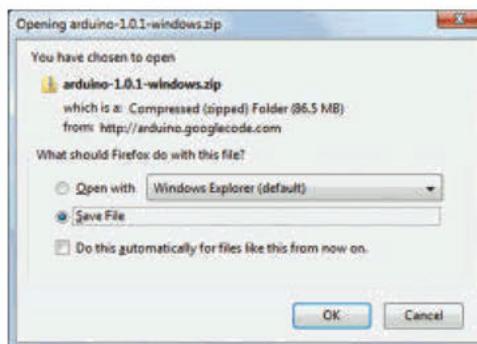


FIGURE 12: Save the file.

that lets you save the file by clicking the Okay button.

The file is placed in the Windows Download Directory as shown in **Figure 13**.

Click on the File menu and select the 'Extract All' item as shown in **Figure 14**.

In **Figure 15**, you see the dialog box that shows where the zip file is currently located. You will want to unzip the file into a directory on your C: drive, so click the Browse button.

Browse to the C: drive as shown in **Figure 16**. Note that in this figure there are several earlier versions of Arduino

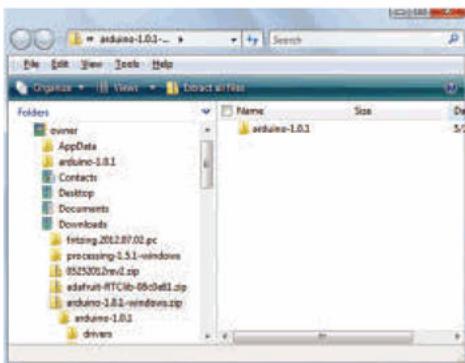


FIGURE 13: Windows downloads directory.

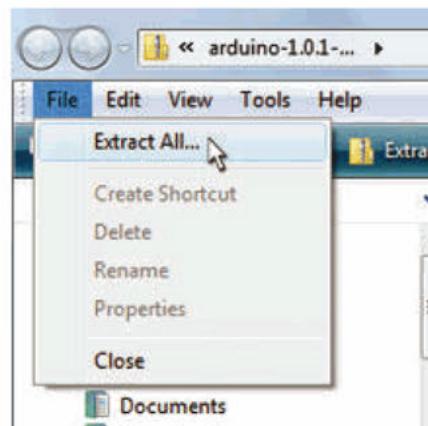


FIGURE 14: Extract All, Step 1.

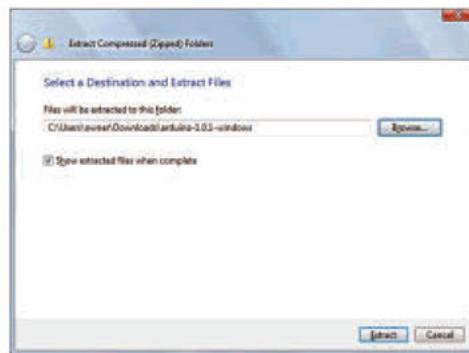


FIGURE 15: Extract All, Step 2.

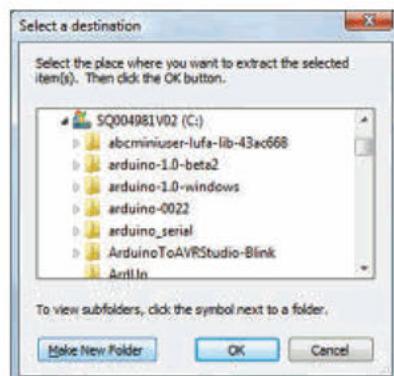


FIGURE 16: Extract All, Step 3.

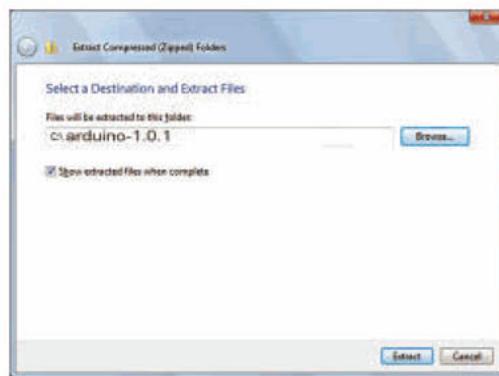


FIGURE 17: Extract All, Step 4.

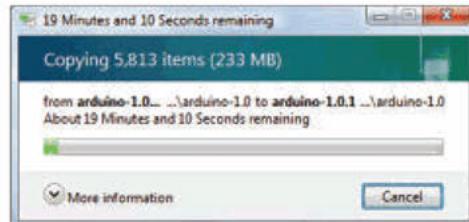


FIGURE 18: Extract All, Step 5.

installed. Just ignore that and click on the ‘Make New Folder.’

- ❑ Name the new folder for the version of Arduino you are using; in **Figure 17**, it is Arduino-1.0.1.
- ❑ Click the ‘Extract’ button. You may want to get comfortable since there are a lot of files that must be unzipped (**Figure 18**).
- ❑ Open the Arduino directory and right-click on the *Arduino.exe* file; select ‘Create Shortcut’ as shown in **Figure 19**.
- ❑ Drag the shortcut to the desktop so you’ll have it handy as shown in **Figure 20**.



FIGURE 20: The official Arduino shortcut.

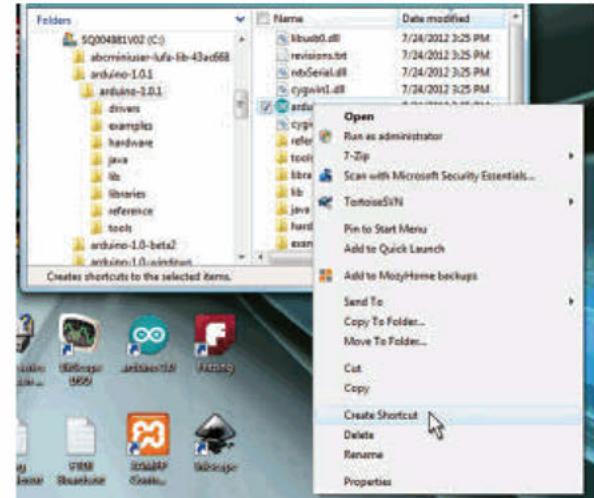


FIGURE 19: Create Shortcut.

Lab 3: The Arduino IDE

The Arduino IDE (Integrated Development Environment) lets you write sketches (also known as programs) that provide instructions telling the Arduino board what to do and how to do it. The IDE has many features that let you select the particular type of Arduino you’ll be using. Here, we’ll be using the Arduino Uno R3, but one of the great things about Arduino is that the code you write for one particular Arduino variant can be easily changed to use with another variant.

There are also many useful example sketches included in the IDE, and looking at these examples is one of the best ways to learn how to use the Arduino. So, let’s get started with the IDE.

Required tools:

A computer with access to the Internet.
The Arduino IDE on your computer.

Estimated time for this lab: 15 minutes

Check off when complete:

- ❑ Open the Arduino IDE by clicking on the shortcut (or *arduino.exe* in the Arduino directory). You’ll see the Security Warning shown in **Figure 21**. Just ignore it and click ‘Run’ since the Arduino IDE is quite secure.
- ❑ You will first see the Arduino Wait panel shown in **Figure 22**.
- ❑ After a brief delay for the application to load, you’ll see the IDE when first opened as shown in **Figure 23**. (Note that it may vary slightly if you are using another version or seeing it on a Mac or Linux machine.)
- ❑ Take a moment to familiarize yourself with this IDE. Note that the top line says *sketch_july24a|Arduino 1.0.1*. This identifies the name of the sketch (the program you’ll write momentarily) and the version of the Arduino you are using. The next line has the menu selections for File, Edit, Sketch, Tools, and Help. We’ll go into greater detail on these in later labs.
- ❑ The next line has six icon buttons shown labeled in **Figures 24** to **29**.
- ❑ **Figure 24** shows the ‘Verify’ icon. After you write a sketch, you’ll click on this icon. This converts the sketch into a form that can be understood by the microcontroller. (This is also known as compiling.)
- ❑ **Figure 25** shows the ‘Upload’ button that is used to transfer the verified sketch to the Arduino.



FIGURE 21: Ignore this security warning.

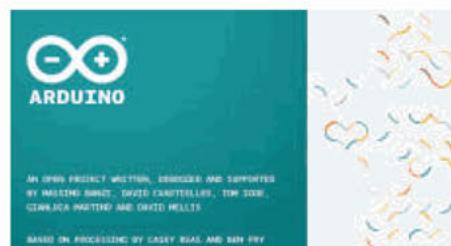


FIGURE 22: Arduino Wait panel.

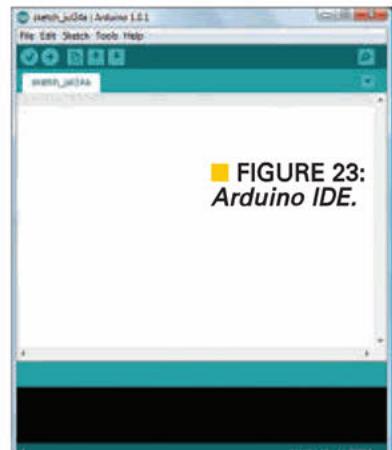


FIGURE 23: Arduino IDE.

- ❑ **Figure 26** shows the ‘New’ button that is used to open a new instance of the Arduino IDE containing a new blank sketch.
- ❑ **Figure 27** shows the ‘Open’ button that you use to locate and open an existing sketch (lets you find an Arduino program on your computer).
- ❑ **Figure 28** shows the ‘Save’ button that you use to save a sketch to your Arduino Sketch directory (or wherever else you want to save it).
- ❑ **Figure 29** shows the ‘Serial Monitor’ button that opens the serial monitor which is an application on your PC that lets you talk to the sketch you’ve uploaded to the Arduino.
- ❑ Now, refer back to **Figure 23** and notice the large white area in the middle with the ‘sketch_jul24a’ tab above it. This area is a text editor that you will use momentarily to write your first sketch.
- ❑ Also note the smaller black area near the bottom of **Figure 23**. This area provides feedback from the verification and upload process. You need to know that the verification and upload processes are done by two

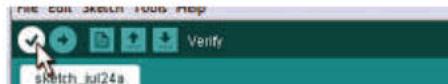


FIGURE 24: Verify button.

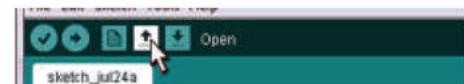


FIGURE 27: Open button.



FIGURE 25: Upload button.



FIGURE 28: Save button.



FIGURE 26: New button.

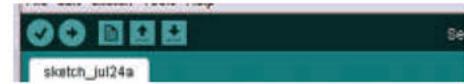


FIGURE 29: Serial Monitor button.

other applications that the Arduino IDE calls. These applications are fairly complex and they provide error messages that may help, but these messages can be very obtuse. We’ll look at this in more detail later.

- ❑ Finally, on the very bottom of **Figure 23** you see a line that identifies the Arduino we are using (Arduino Uno) and the communication port for the PC (COM4). The COM port will probably be different on your setup.

Lab 4: Compile a Sketch to Blink an LED

You can see the pin 13 LED back in **Figure 1**. The Arduino senses and actuates through its pins. It has a built-in LED on pin 13 that we can blink with a simple sketch.

Required tools:

A computer with access to the Internet.
The Arduino IDE on your computer.
An Arduino board, preferably the Uno R3.

Estimated time for this lab:

20 minutes

Check off when complete:

- ❑ Place your cursor in the sketch editor and type the following:

```
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

This is shown in **Figure 30**.

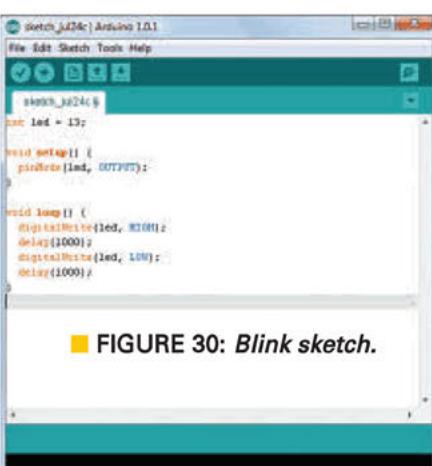


FIGURE 30: Blink sketch.

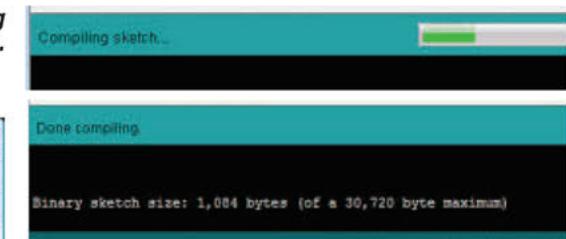


FIGURE 31: Compiling the sketch.

FIGURE 32: Done compiling.



FIGURE 33: Error 'dela' not declared.

Lab 5: Upload the Blink LED Sketch

Now that we have the verified sketch, we can upload it to the Arduino to blink the LED attached to pin 13.

Required tools:

A computer with access to the Internet.
The Arduino IDE on your computer.
An Arduino board, preferably the Uno R3.

Estimated time for this lab:

15 minutes

Check off when complete:

- Hook a USB cable between your PC and the Arduino as shown in **Figures 34** and **35**.
- Click on the Tools menu item and select Serial Port, then click on the checkbox for the port as shown in **Figure 36**.
- If more than one serial port is indicated, then you should close the Tools/Serial Port item, unplug your Arduino, then reopen the serial port to see which port has gone missing. Then, plug your Arduino back in and select that port as shown in **Figure 37**.
- Click on the Upload button in the Arduino IDE and watch the TX and RX lights on the Arduino board. They should rapidly blink for a few moments indicating that the PC and the Arduino are talking to each other so you can upload the program.
- Now, look at the pin 13 LED as shown in **Figure 38**; it should be blinking on and off once a second. Congratulations! You've programmed an Arduino!
- Open the File menu and click on 'Save As' to save this sketch as MyBlinky.
- Open the File menu and click on 'Sketchbook.' Note that your program has been saved.



FIGURE 34: USB cable attached to the PC.



FIGURE 35: USB cable attached to the Arduino.

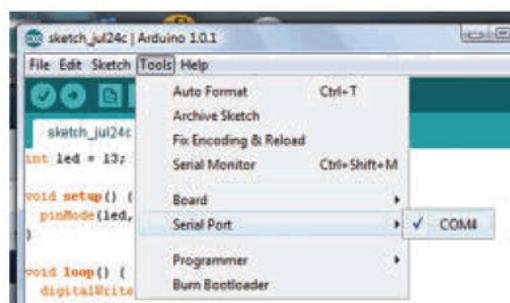


FIGURE 36: Tools, Serial Port.

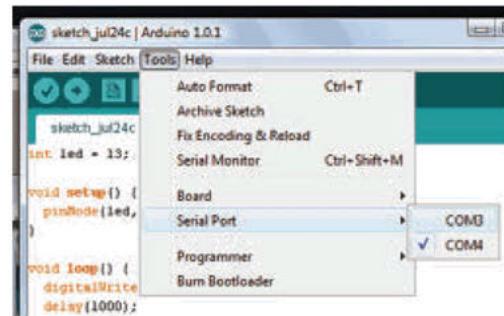
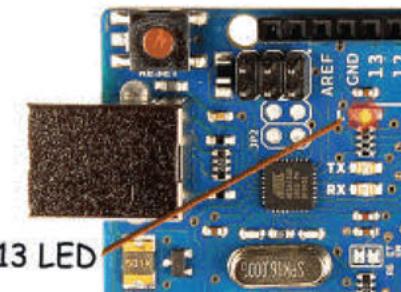


FIGURE 37: Multiple serial ports.



Pin 13 LED

FIGURE 38: The pin 13 LED blinking.

Lab 6: Using an Example Program

The Blink sketch that you just typed into the Arduino editor is taken from an example that you can load from the Examples item under the File menu item. Yes, you didn't really need to type in that program, but the exercise gave you some insight into both how easy it is to write a program and how easy it is to make mistakes.

Required tools:

A computer with access to the Internet.
The Arduino IDE on your computer.

An Arduino board, preferably the Uno R3.

Estimated time for this lab:

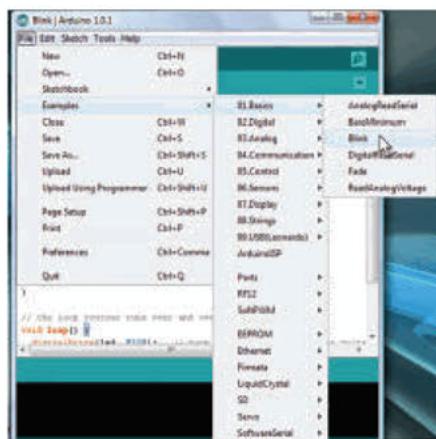
10 minutes

Check off when complete:

- Click on the File menu item and select Examples – Basics – Blink as shown in **Figure 39**.
- Read the Blink program shown in **Figure 40** and pay special attention to all the comments (gray words following // or bracketed by /* and */). When you typed in the program, you weren't asked to add these comments. In the future, you'll want to be very liberal with your

comments since they will help others understand your code and will help you in six months when you've forgotten why you did something the way you did. Comments are a critical part of project documentation.

FIGURE 39: Select Blink example.



```
File Edit Sketch Tools Help
Blink
/*
Blink
TURNS ON AN LED ON FOR ONE SECOND, THEN OFF FOR ONE SECOND, REPEATING THIS CYCLE FOREVER.
*/
// PIN 13 HAS AN LED CONNECTED ON MOST ARDUINO BOARDS.
// GIVE IT A NAME.
int led = 13;

// THE SETUP ROUTINE RUNS WHEN YOU PRESS RESET.
void setup() {
  // INITIALIZE THE DIGITAL PIN AS AN OUTPUT.
  pinMode(led, OUTPUT);
}

// THE LOOP ROUTINE RUNS OVER AND OVER AGAIN FOREVER.
void loop() {
  digitalWrite(led, HIGH); // TURN THE LED ON (HIGH = TRUE).
  delay(1000); // WAIT ONE SECOND.
  digitalWrite(led, LOW); // TURN THE LED OFF (LOW = FALSE).
  delay(1000); // WAIT ONE SECOND.
}
```

FIGURE 40: The Blink example program.

Lab 7: Using the Reference

The Arduino IDE has a built-in reference book that provides explanations for most of the Arduino's features. This is an extremely valuable tool to help you learn about the Arduino. Let's see how to use it.

Required tools:

A computer with access to the Internet. The Arduino IDE on your computer. An Arduino board, preferably the Uno R3.

Estimated time for this lab: 10 minutes

Check off when complete:

- Open the Help menu and click on 'Reference' as shown in **Figure 41**.
- The Arduino Reference opens in a browser as shown in **Figure 42**.
- Take a few minutes to look at the various sections, then click on a few links to see how to navigate among the various terms and their web pages.
- Find 'Terms' in the Reference area.
- Arduino has a very helpful feature that allows you to highlight a word and then see if you can find it automatically in the Reference.
- In the Blinky program, highlight the term `pinMode` as shown in **Figure 43**.
- Open the Help menu and click on 'Find Reference' as shown in **Figure 43**.
- The Arduino Reference opens on the entry for the term in a browser as shown in **Figure 44**.
- Take a few minutes to look at the various sections, then click on a few links to see how to navigate among the various terms and their web pages.

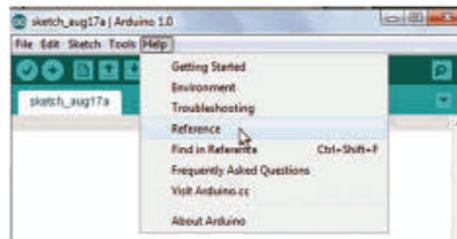


FIGURE 41: Select Reference.



FIGURE 42: Reference in browser.

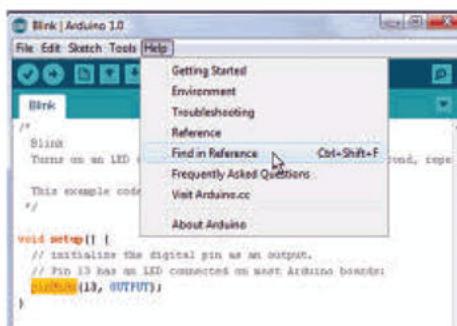


FIGURE 43: Find in Reference.



FIGURE 44: pinMode found in Reference.

Next Month

In Chapter 2 of Arduino 101, we will start covering some of the basics of electronics that are needed to build projects with the Arduino. **NV**

SDS7102 100MHz 1GSa/s 10Mpt 2-Channel Oscilloscope



\$429

Deep Memory - Battery Option!

The Owon SDS7102 is a 100MHz bandwidth scope with huge 10Mpt memory for fast signal investigations. Battery option for use away from 110V - great for field work!

"I never expected a high end scope for around \$400 but Owon delivers more features than I anticipated." -N.

"For the price this scope cannot be beat!" -D.

Your choice!

2-Channel Digital Storage Oscilloscopes with High-End Features:

High-performance, economic digital oscilloscopes for education, training, production line, and R&D. 8-inch color 800x600 TFT_LCD screen, AutoScale, function to simplify use. DB-15 VGA port shows screen display on external monitors or video projectors - ideal for teaching. USB flash disk storage, Pass/-Fail, LAN for remote measurements. Sophisticated triggering includes: Edge, Pulse, Video, Slope. Waveform recording and replay. Comes with probes, USB cable, probe adjust pen, software CD, power adapter, and user manual.



Saelig^{com}
unique electronics
www.saelig.com 888-7SAELIG

SDS5032E 30MHz 250MSa/s 2-Channel Oscilloscope



\$289

Lowest Cost Scope!

The Owon SDS5032E is an economical 2-channel 30MHz 250MSa/s digital storage oscilloscope with 10Kpt memory. Excellent starter scope!

"Great scope for the price. Love this scope, especially the clear full color 800x600 screen." -A.

"...nice variety of features and unmatched for price." -Z.

A New Airframe Design for Near Spacecraft: Part 2 — Dealing with the Chaos

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/january2014_NearSpace.

Building a new airframe is only part of creating a near spacecraft. After all, what good is an empty box? This time, I'll begin discussing how I modified the interior of my newest airframe design. I added these modifications to the old design to prevent the valuable contents of the airframe from bouncing around during descent or — even worse — leaving the airframe all together!

Post Burst Chaos — Just Another Day in Near Space

As I said last time, my second near spaceflight in July 2013 tumbled pretty badly during its descent from near space. So bad, in fact, that we lost track of its position and relied on a local farmer to find it for us.

Mike Manes (W5VSI) of the Edge of Space Sciences (EOSS) in Colorado describes the initial descent

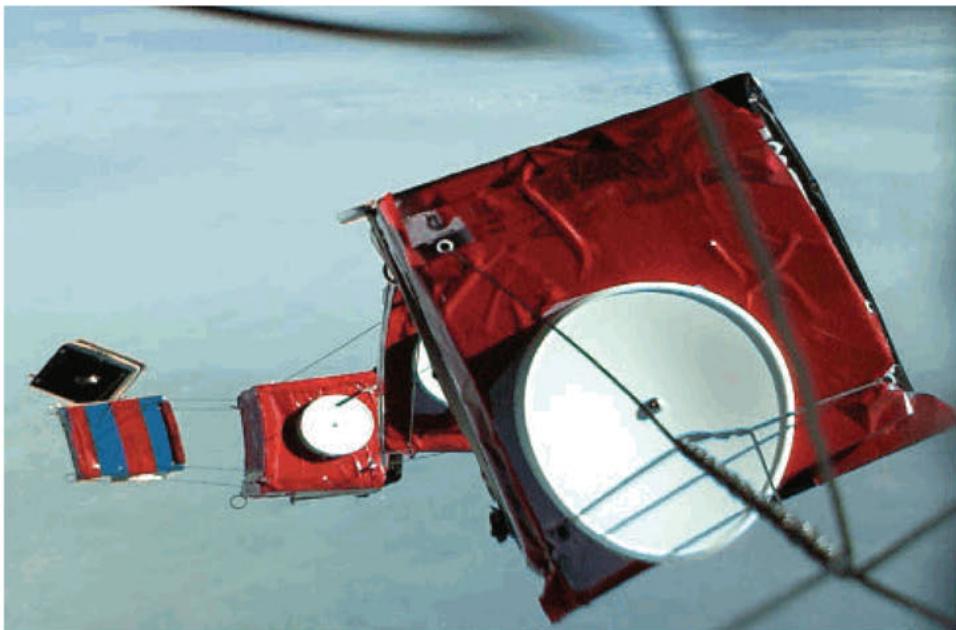


Figure 1. The chain of BalloonSats in a near spacecraft normally are pointing downward to earth. During descent, however, conditions get very turbulent and the BalloonSats are likely to swing like a chaotic pendulum. There are instances of the BalloonSats swinging higher than the parachute at the top of the near spacecraft.

of a near spacecraft as post burst chaos (PBC). If you watch a video of a near spacecraft's descent, you'll understand why. Even though the parachute is opened (it opens within one second of balloon burst), the high altitude descent of a payload can be very traumatic.

There are two major sources for this chaos: one from the top of the near spacecraft and one from the bottom. The parachute — which we design for low speed descents at sea level — snaps open and closed in a quick and random sequence during the low pressure/high speed portion of the descent.

The fact that balloon fragments

remain attached to the apex of the parachute makes the situation even worse. The leftover pound or so of latex balloon connects to the parachute's apex with 15 to 20 feet of line. This creates a rapidly swinging torque that tugs the parachute in ever-changing directions.

At the other end of the near spacecraft, we have the payload modules. The near spacecraft is typically a long string of modules (sometimes called the flight train) which performs functions like tracking (there may be a backup tracker in the train) and data collection (these can be student-built

BalloonSats). The modules are separated from each other by at least a foot to reduce potential collisions between them.

There can be up to a dozen modules linked together with one or more strands of nylon cord called link lines. Together, they form a long pendulum of individual modules which can result in a lot of swinging and tumbling during descent.

The oscillation of the modules is not regular or very predictable; in part, because of the aerodynamic forces acting on the individual modules that have unique weights and sizes.

When we combine the snapping of the parachute and the swinging of the flight train, there's a lot of changing linear and rotating forces acting on the near spacecraft. We can see what this is like for the near spacecraft in the accelerometer data displayed in **Figure 2**.

There are two techniques to limit the disruptive forces of descent. First, you can place a dense aerodynamic object at the bottom of the near spacecraft. This "aerospike" helps by keeping the stack of modules aligned with the ground by the tension it creates on the link lines. Keeping modules more vertical reduces the opportunities for modules to swing around like independent pendulums.

Second is to cut the entire load line (the cord attaching the balloon to the parachute apex); the balloon remains away from the parachute as soon as burst occurs. Without torque from the balloon tugging on the parachute, there's less force pulling the parachute closed and over on its side.

The force acting on the parachute's apex can still be significant when only ounces of balloon remain after burst. That's because the balloon attaches to the parachute apex by a 10 to 20 foot long load line. So, it's important for the separation mechanism to release the load line also. EOSS has a great mechanism for separating the balloon

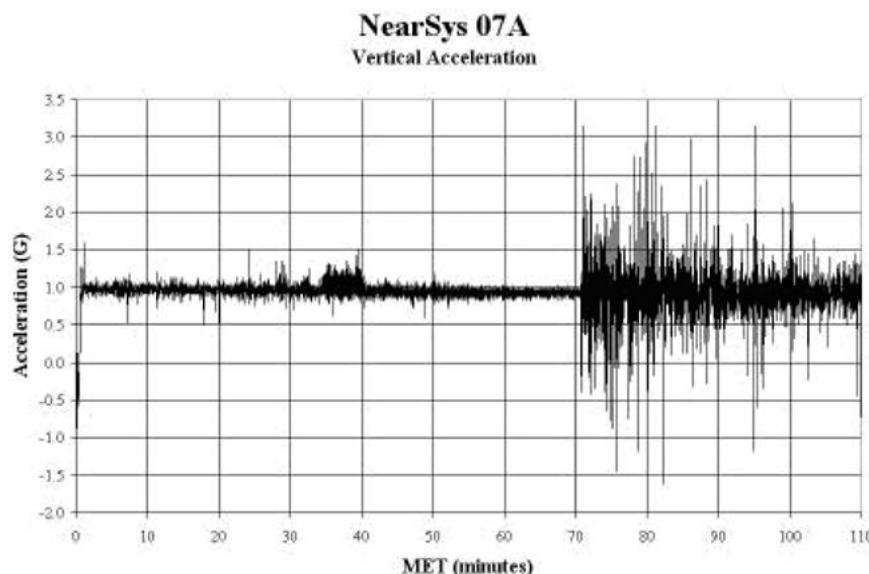


Figure 2. This shows some of the worst aspects of a descent from near space. At balloon burst (71 minutes into the mission), the acceleration experienced by the near spacecraft varies from -1.5 gs to +3.0 gs in a random pattern. This creates a lot of strain on the avionics inside the airframe.

from the parachute and I encourage you to check it out some time.

Dealing With Chaos

I'm still trying to produce an aerospike module and an acceptable balloon release mechanism. Until then, I am using construction methods that protect the avionics

from the effects of PBC. The need to protect the avionics was brought to my attention in near space mission NearSys 10G.

Shortly after balloon burst, one of the trackers stopped producing output. Fortunately, the second tracker continued transmitting position reports every minute so we were able to locate the near

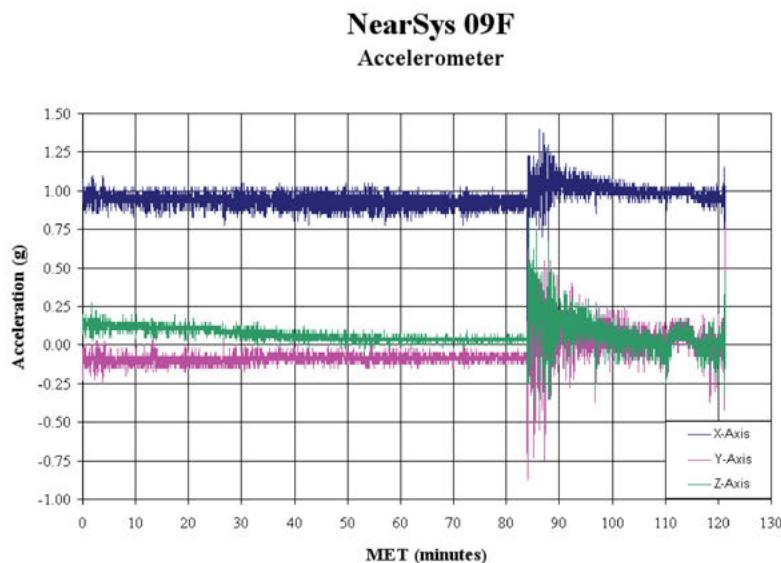


Figure 3. Here, we see that the situation is even worse during the initial descent. The acceleration experienced by the near spacecraft varies in all three dimensions and in random patterns.



Figure 4. Mission NearSys 10G made it abundantly clear that post burst chaos is a serious problem. We were very surprised by what we discovered after extracting this near spacecraft from the tree it landed in.

spacecraft once it landed.

Upon arriving at its position, we discovered that it had landed in a tall tree. It took a couple hours to extract it and when we finally did, we found out just why the first tracker stopped producing position reports. The entire module was empty.

There were no avionics, batteries,

or cameras inside. Descent had spun and shaken the module so badly that it tore off its hatch. Once the hatch — which was the only thing holding the avionics inside the airframe — came lose, the valuable contents of the airframe shot out. Somewhere in Kansas, there's a farmer who found a nice GPS receiver, miniature video camera, radio transmitter, and a programmable flight computer in his wheat field.

Up until this time, I used Velcro™ straps to seal the hatch of the airframe. For over 80 missions, that was enough. Therefore, when I created my newest airframe design, I resolved to strengthen the hatch closure and bolt every bit of the avionics inside the airframe.

Avionics Pallet Changes

Let's begin with how I now secure the avionics. I always attach my avionics to a sheet of Coroplast. Coroplast is a 1/4" thick sheet of corrugated plastic made from polyethylene. I like the material because it's stiff and lightweight. I attach the printed circuit board of

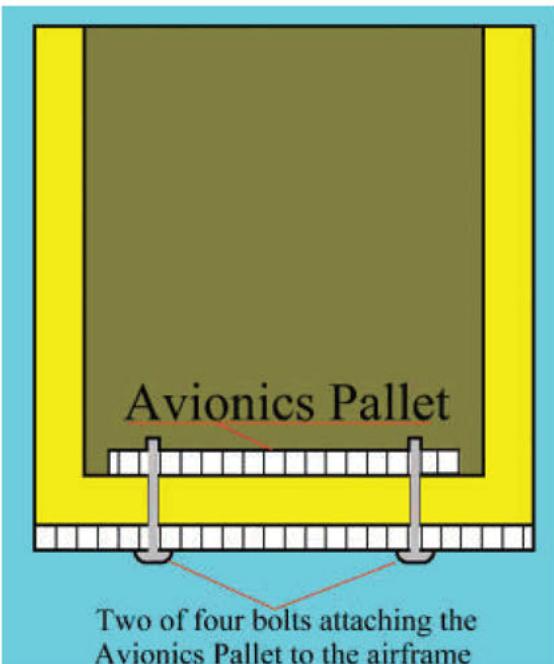


Figure 5. This cross section illustrates how the bolts used to attach the impact plate and straps to the airframe are also used to bolt the avionics pallet to the airframe. My hand can reach inside the airframe and screw on wing nuts to the bolts.

the flight computer to the Coroplast using nylon zip ties, then place the avionics pallet in the bottom of the airframe.

In my old design, I filled the remaining open volume with Styrofoam packing peanuts. This is fine as long as the hatch remains sealed. The packing peanuts help to prevent the avionics pallet from shifting around.

However, as I saw in 2010, when PBC rips the hatch off, there's nothing to prevent the packing peanuts and avionics pallet from departing the airframe. Therefore — in my new airframe — the pallet is bolted to the bottom of the airframe. It doesn't take much strength to keep the avionics pallet in place. In fact, I use the same bolts that attach the straps to the airframe — I just use longer bolts.

I use wing nuts rather than nylocks to secure the avionics pallet to the bolts because it's difficult to reach into the airframe with a wrench or pair of pliers. Since wing nuts aren't nylocks, it's important to check the wing nuts for tightness prior to each flight.

Even with the avionics pallet secured to the airframe with bolts and wing nuts, I still fill the remaining volume of the airframe with packing peanuts. Now, they're only used to spread out the forces the avionics are subjected to and pass them to the entire airframe.

Next up is the battery/GPS pallet at the top of the airframe and the hatch to button this up. I'll go over this next time, and explain how I designed them to stay in place through the PBC.

Don't forget the Great Plains Super Launch takes place in Hutchinson, KS this year in June. If you want to learn the ropes of near space flights, this is the event to attend. You can find out more at www.superlaunch.org.

Onwards and Upwards,
Your near space guide **NV**

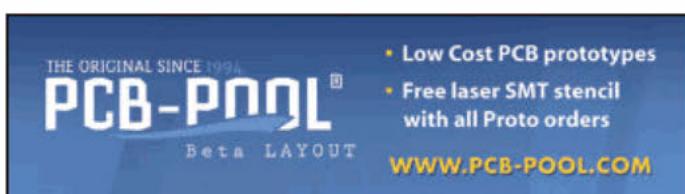
ELECTRONET

For the ElectroNet
online, go to
www.nutsvolts.com
click **Electro-Net**



USB Add USB to your next project--
It's easier than you might think!
DLP Design
USB-FIFO • USB-UART • USB/Microcontroller Boards
RFID Readers • Design/Manufacturing Services Available
Absolutely NO driver software development required!
www.dlpdesign.com

Got electronics?
www.nutsvolts.com



THEY SAY THE FUTURE DOESN'T FIT IN A BOX.

Tuesday, January 7
through Friday, January 10, 2014
Las Vegas, Nevada • CESweb.org • #CES2014

**GOOD THING LAS VEGAS
IS FAR FROM SQUARE.**

Over four days, those who shape the future gather in a city built on reinvention. Here, brands, markets and economies converge in what's far more than a tradeshow. And in 2014, there's more opportunity than ever to connect with those who matter. The only question now, why wouldn't you fit it in your future?

Register today at CESweb.org.



THE GLOBAL STAGE FOR INNOVATION

PRODUCED BY  CEA

Utilizing Embedded Web Development

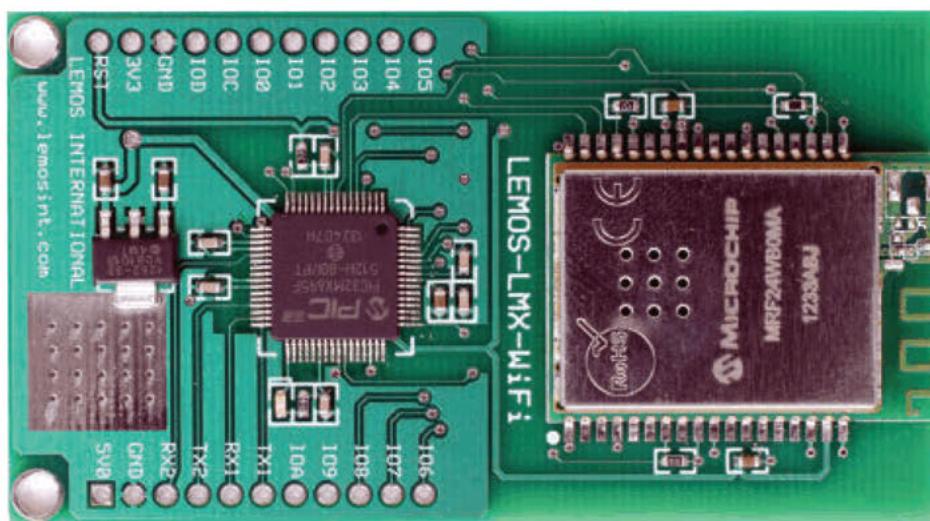
Going to school has always been about studying and learning about something that someone did in the past. Even with today's instant communication facilities, it has not changed. We are all still learning from someone else's previous successful (or unsuccessful) exploits.

This month, we will continue the learning cycle by examining some rather obvious firmware generation documentation. In the meantime, we will also examine a new piece of Wi-Fi hardware.

What Front Panel?

I grew up with chicken head knobs and toggle switches. Today, those famous radio and instrumentation controls are being rapidly replaced by web browser views and touch screens. To be truly interactive, the new-fangled electronic interfaces need support from both the device firmware and the interface medium. It's not

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/index.php?/magazine/article/january2014_DesignCycle.



■ Photo 1. Don't be fooled by the LMX Wi-Fi module's lack of component complexity. This shot contains a PIC32MX695F512H that is more than capable of housing the complete Microchip TCP/IP stack and executing a complex embedded Wi-Fi application.

enough to just write some microcontroller firmware and throw it at a web page. The inherent capabilities of a web page must be utilized in conjunction with the host device's application firmware.

Thanks to some tools of the trade, we as embedded designers don't have to burn the midnight oil mulling over HTML, XML, and JavaScript. Let's take a layman's tour of a typical embedded application that takes advantage of the power of the Web.

The Hardware

Photo 1 details a Microchip PIC32MX695F512H 32-bit microcontroller coupled with their MRF24WB0MA Wi-Fi radio. The idea behind this Lemos LMX-WiFi module used here is modularity. Everything needed to support the

MRF24WB0MA is found in the
PIC32MX695F512H.

The PIC is endowed with 512K of program Flash. It is equally capable on the SRAM front, incorporating 128K of readily available memory space. When used in a web-based embedded application, the LMX Wi-Fi module can support a full-blown TCP/IP stack.

Note the absence of external EEPROM memory which is normally present to hold the web pages and associated data. In that the PIC32 program Flash is capable of being written by command, the web pages and supporting data can be stored on-chip instead of off-chip on external EEPROM.

Note also that the LMX Wi-Fi does not pull each and every I/O line out to a header position. That's because the PIC32's I/O pins can be

multiplexed. Enough of the select I/O pins have been terminated at headers to support I^C, SPI, and UART portals. In the case of the UART, there is a pair of portals. In addition, the LMX Wi-Fi exposes analog-to-digital converter (ADC) pins. A number of additional PIC32 general-purpose I/O pins are also made available to the embedded designer.

The part count for the LMX Wi-Fi is kept at a minimum by employing the services of the PIC32's internal processor clock. An onboard LDO voltage regulator allows an external supply that lies between +3.7 VDC and +5.0 VDC to power the Wi-Fi module.

The PIC32's versatile I/O layout allows the LMX Wi-Fi module to be programmed to perform many functions that are hard-coded into other Wi-Fi units. The LMX module can be programmed to act as a serial-to-WiFi device, a remote sensor platform, and a remote control device. The sensor and control applications are extendable to the Internet. Basically, if an external device can use one of the LMX Wi-Fi's SPI or I^C portals, that device has the capability to go worldwide via the 'net.

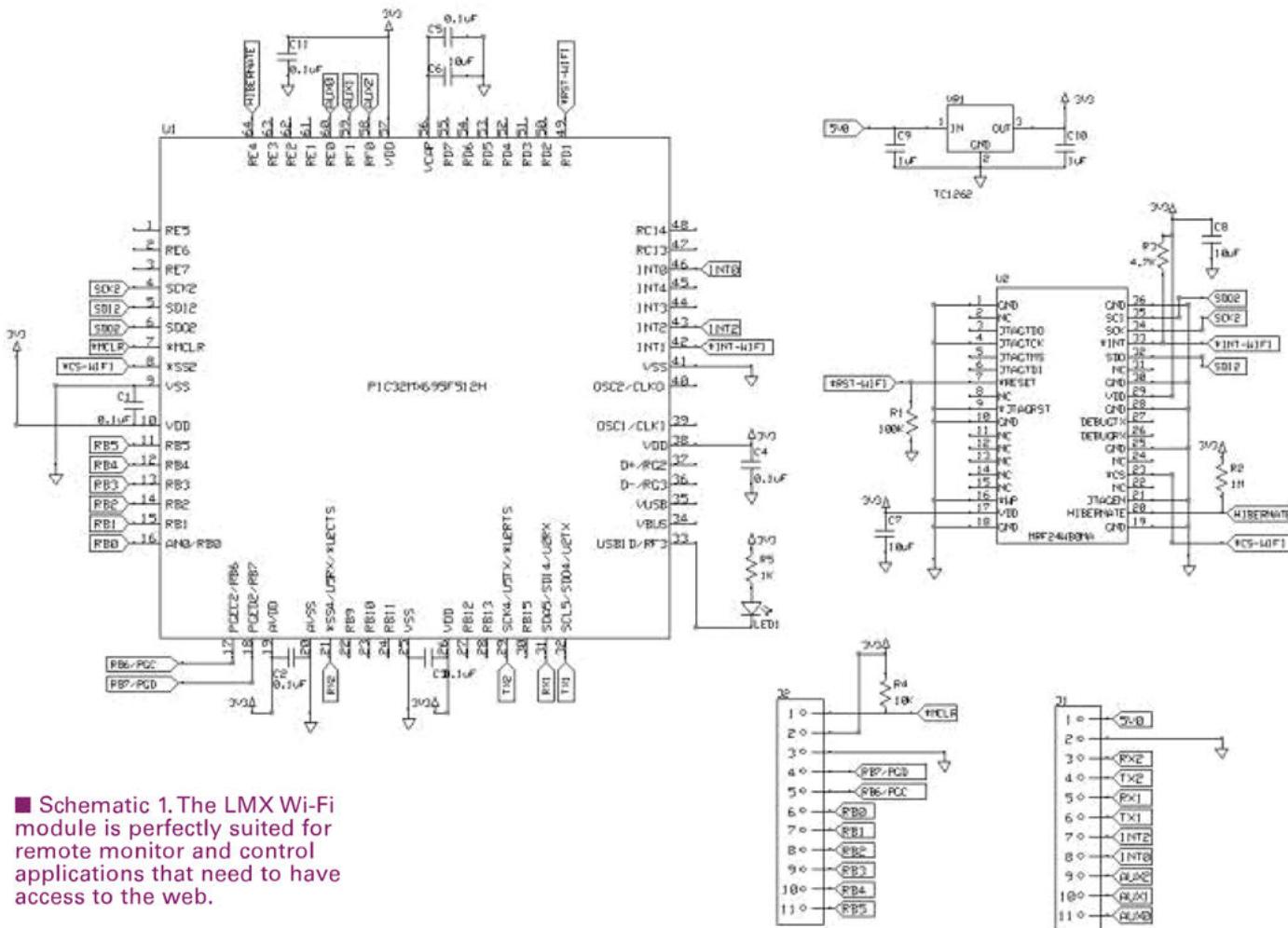
For debugging and programming purposes, the LMX

module lays out a PICkit 3 compatible set of header pads which (in correct order) includes the RST, 3V3, GND, IOD, and IOC header positions. This arrangement allows easy access to the programming and debugging resources offered by MPLAB X and the PICkit 3. The physicals that I have described here are graphically detailed in **Schematic 1**.

Embedded Web Server Mechanics

The web pages that are served by the Wi-Fi module are processed by the Microchip MPFS2 Utility. If the web pages are to be stored in external EEPROM, the MPFS2 Utility spits out a binary file that contains the web page content. If the web pages are to be contained within program Flash, the MPFS2 Utility creates an *HTTPPrint.h* file and an *MPFS2 Image.c* file. The contents of the *HTTPPrint.h* file include all of the references and indices to the callback functions. The *CustomHTTPApp.c* file is the heart of the web application.

The programmer must manually build a callback function for each of the references in the *CustomHTTPApp.c* file. The *Image.c* file contains a



■ Schematic 1. The LMX Wi-Fi module is perfectly suited for remote monitor and control applications that need to have access to the web.

translated image of the web pages, JavaScript files, icon files, XML files, CSS style sheet files, CGI files, and graphic file images. The MPFS2 Utility files along with the *CustomHTTPApp.c* and any application-specific files are all assembled in an MPLAB X project. The compiled image is then programmed into the target PIC's program Flash. The *HTTPPrint.h* file is updated each time the MPFS2 Utility is run after changes have been made to the callback functions.

The HTTP2 server is the overall boss of the web application. Commands in the HTML code trigger callback functions that are executed by the server. The callback functions write data to the web page.

Embedded Cosmetology

One of the many advantages to the LMX Wi-Fi module's design is that Microchip has provided a wealth of ready-to-run MRF24J40MA examples. I've studied them in detail, and I'm ready to show you how to put up a simple interactive embedded web page using the LMX Wi-Fi module. Instead of writing every byte from scratch, we will reuse and customize the JavaScript and CSS (Cascading Style Sheets) that are included with the MRF24J40MA demos.

The LMX Wi-Fi module sports its own blue LED which is driven by the PIC32 I/O port RF3. Our goal is to illuminate and alternately extinguish the LED by simply clicking on an LED symbol painted on the web page. The web page logical LED color will follow the physical LED, which is blue for illuminated and gray for extinguished. As you can see in **Schematic 1**, the module's blue LED illuminates when a logical low is applied to I/O pin RF3.

Building web apps by hand writing HTML is not one of my favorite things. So, we'll employ the services of Dreamweaver to help us generate the necessary HTML tags and content. Our minimal web page can be seen in **Screenshot 1**. Dreamweaver is a WYSIWYG (What You See Is What You Get) web development application. So, I simply typed in the text and Dreamweaver generated the HTML source.

Dreamweaver also allows the insertion of objects such as pictures and graphics. That explains the authentic *Nuts & Volts* image. The circular LED graphic was generated using the HTML *•* symbol entity. The variables enclosed in *"~* characters are directly connected to variables that are being processed in the web server code. For instance, *~ssid~* will be replaced by the actual connected WLAN SSID. These *"~surrounded"* variables are called dynamic variables.

The web page you see in **Screenshot 1** is called *index.htm*. In this scenario, the *index.htm* page is always displayed at power-up or reset. Coding of the *index.htm* page involves melding HTML tags and JavaScript statements. HTML is in a huge way responsible for what



Design Cycle

Current WLAN Connection:

- SSID name: *~ssid~*
- Security: *~curPrivacy~*
- WLAN type: *~wlan~*
- IP Address: *~ipaddr~*

■ **Screenshot 1.** Dreamweaver is a WYSIWYG web development tool. Basically, you simply type in what you want and Dreamweaver generates the HTML source.

you see on a web page. JavaScript helps by providing some of the muscle that makes things "happen" there. JavaScript commands are executed by the web browser under control of the HTTP2 server component of the Microchip TCP/IP stack. For instance, this line of HTML puts up the *Nuts & Volts* graphic:

```
<div id="logo_banner"></div>
```

In the HTML statement that follows, the HTML event attribute *onclick* kicks off a JavaScript function. The JavaScript function (*newAJAXCommand*) is aimed at *ledblue* which happens to be mapped to the Wi-Fi module's onboard blue LED:

```
<a id="ledblue" onclick="newAJAXCommand('led.cgi?lednumber=0');">&bull;</a>
```

When the web page LED is clicked, JavaScript invokes the *newAJAXCommand* function. The *newAJAXCommand* function in conjunction with the *led.cgi* file returns a value that is associated with the argument *lednumber*. The returned data is used to determine the ultimate state of the web page LED (*•*) and the physical LED. The HTTP2 portion of the TCP/IP takes care of the mechanics of data movement behind the scenes. The returned data ends up in *curHTTP.data*. All we have to do is parse *curHTTP.data* for the data value associated with the *lednumber* argument:

```
*****  
// If it's the led.cgi LED update file  
*****  
if (!memcmp(ppgm2ram(filename, "led.cgi", 8))  
{  
    ptr = HTTPGetROMArg(curHTTP.data, (ROM  
BYTE *)"lednumber");  
  
    if (*ptr == '0')  
    {  
        ledF3_INV(); //toggle the physical LED  
    }  
}
```

An ASCII '0' flips the state of the physical blue LED.

If you're wondering about AJAX, it is a method that combines JavaScript and XML to allow portions of a web page to be modified without having to reload the entire web page. AJAX is short for Asynchronous JavaScript and XML. AJAX is not a programming language. It simply uses existing web page access techniques to work its magic. AJAX is used in this project to speed up the refresh rate of the LED web page object.

The heart of the *newAJAXCommand* is the XMLHttpRequest object. Invoking XMLHttpRequest allows the programmer to:

- Update a web page without reloading it.
- Request data from a web page after it has been loaded.
- Receive data from a web page after it has been loaded.
- Send data to a server in the background.

Like AJAX, XML is not a programming language. XML is used to format and store the data that is being thrown about. Here's what the XML looks like for the Wi-Fi module's blue LED:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <ledblue>~ledxml(0)~</ledblue>
</response>
```



Design Cycle

Current WLAN Connection:

- SSID name: LEMOS_591a
- Security: None
- WLAN type: AdHoc (IBSS)
- IP Address: 169.254.1.1

LED



Design Cycle

Current WLAN Connection:

- SSID name: LEMOS_591a
- Security: None
- WLAN type: AdHoc (IBSS)
- IP Address: 169.254.1.1

LED



■ Screenshot 2. Pretty cool stuff! Just think. If we can monitor and control an LED using embedded web services, we can use these concepts to monitor/control anything.

The *ledxml(0)* data element is represented by the XML tags <ledblue> and </ledblue>. There are no predetermined tags in XML. Therefore, we can create tags that make the XML easy to read and understand.

We've already toggled the blue LED that is soldered to the module's circuit board. To update the LED on the web page, we must again turn to XML, JavaScript, and the *led.cgi* file.

Clicking on the web page causes the HTTP2 server component to order the execution of a callback function. The callback order comes from encountering the ~*ledxml(0)*~ dynamic variable that makes up the entire logical contents of the *led.cgi* file. The *HTTPPrint_ledxml* callback function is executed by the PIC32 and sends the current state of the physical blue LED to the web browser:

```
/*
// LMX WiFi Blue LED Hardware Assignments
//
#define ledF3_TRIS          (TRISFbits.TRISF3)
#define ledF3_IO             (LATFbits.LATF3)
#define ledF3_ON()           LATFCLR = BIT_3;
#define ledF3_OFF()          LATFSET = BIT_3;
#define ledF3_INV()          LATFINV = BIT_3;

void HTTPPrint_ledxml(WORD ledState)
{
    ledState = ledF3_IO;

    // Print the output
    TCPPut(sktHTTP, (ledState?'0':'1'));
    return;
}
```

Now that the physical LED state is known by the web page, we can invoke some JavaScript code. This JavaScript code snippet flips the state of the logical LED on the web page relative to the LED state data that was received:

```
if(getXMLValue(xmlData, 'ledblue') == '1')
    document.getElementById('ledblue')
        .style.color = "blue";
else
    document.getElementById('ledblue')
        .style.color = '#ddd';
```

Here's how it all fits together. The current logical state of the PIC32's RF3 I/O pin is determined and sent to the web browser by the *HTTPPrint_ledxml* callback function. The LMX module's blue LED is illuminated by forcing the I/O pin RF3 logically low. Thus, when the physical blue LED is illuminated, the *HTTPPrint_ledxml* callback function sends an ASCII '1' (0x31).

Conversely, a logical high on RF3 says that the physical blue LED is extinguished and the *HTTPPrint_ledxml* callback function sends an ASCII '0' (0x30). The transmitted XML data is picked up by the web browser and processed in the JavaScript code snippet. An ASCII 1 paints the web page LED (•) blue. Otherwise, the web page LED is painted gray if an ASCII '0' is received. I captured the logical LED transitions in **Screenshot 2**.

Where Did Those Dynamic Variables Go?

It appears that the dynamic variables shown in **Screenshot 1** have been replaced. The logic behind the dynamic variables is the same logic that controls the logical and physical LEDs. Recall that the *HTTPPrint.h* file knows all about each and every dynamic variable. The dynamic variables in our web application are all indexed and tied to a callback function in the *CustomHTTPApp.c* file.

When a dynamic variable is encountered, it triggers the HTTP2 server component to force the execution of the associated callback function. Let's follow the dynamic variable `~ssid~` as it morphs into *LEMSO 591a*.

The `TCPConfig.h` file holds many of the configurable data elements that define the network. Let's give our Wi-Fi module a new host name:

```
#define MY_DEFAULT_HOST_NAME "LEMOS"
```

A default IP address may be a good idea as we can't be sure of what type of network the LMX Wi-Fi Module will be a part of:

```
#define MY_DEFAULT_IP_ADDR_BYTE1 (169ul)
#define MY_DEFAULT_IP_ADDR_BYTE2 (254ul)
#define MY_DEFAULT_IP_ADDR_BYTE3 (1ul)
#define MY_DEFAULT_IP_ADDR_BYTE4 (1ul)
```

We find the following statement in the `WF_Config.c` file:

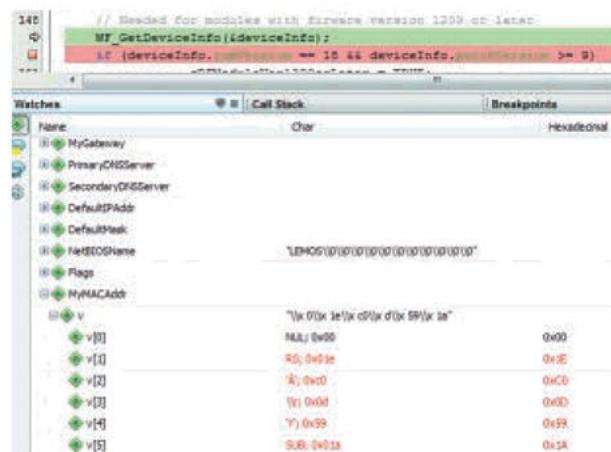
```
/* Append Last 4 digits of MAC address
 * to SSID */
sprintf((char *)CPElements.ssid,"LEMOS_%02x%02x",
        AppConfig.MyMACAddr.v[4],
AppConfig.MyMACAddr.v[5]);
```

CPElements is a C structure that allocates buffer space for the network SSID (*CPElements.ssid*). The *sprint* command loads the *CPElements.ssid* memory area with:

LEMSOS_591a
Where: 59 = AppConfig.MyMACAddr.v[4]
1A = AppConfig.MyMACAddr.v[5]

Screenshot 3 shows us where the `MyMACAddr` data originates. A call to `WF_GetDeviceInfo` loads the `AppConfig` structure with a number of network definitions including the MRF24J40MA's MAC address. The `index.htm` HTML source that displays the network information looks like this:

```
<h2>Design Cycle</h2>
<p>Current WLAN Connection:</p>
<ul>
  <li>SSID name: <b>~ssid~</b></li>
  <li>Security: <b>~curPrivacy~</b></li>
  <li>WLAN type: <b>~wlan~</b></li>
```



- Screenshot 3. This is a snapshot of the MPLAB X debug window. I've carved out the MyMACAddr section of the AppConfig structure.

```
<li>IP Address: <b>~ipaddr~</b></li>
</ul>
```

The dynamic variable `~ssid~` is encountered and HTTP2 orders the following associated callback function:

```
// =====
// = Sends the new/current SSID setting
// =====
void HTTPPrint_ssid(void)
{
    TCPPutArray(sktHTTP, CPElements.ssid,
    CPElements ssidLength);
}
```

With that, let's move on and track down the `~curPrivacy~` substitution. Take a look at **Screenshot 4**. As with the SSID, the `~curPrivacy~` dynamic variable forces HTTP2 to issue a callback request. The `CPElements.securityType` field in **Screenshot 4** reveals a value of `0x00`. The `HTTPPing_curPrivacy` callback follows and it is obvious to the most casual observer what the outcome will be:

```
// ======  
// = Sends the Current Security Setting =  
// ======  
void HTTPPrint_curPrivacy(void)  
{
```

- **Screenshot 4.** This capture exposes the entire CPEElements structure. The securityType value is shown as 0x00. You can see the CPEElements.ssid entry we tracked down earlier.

Lemos International
LMX Wi-Fi Module
www.lemosint.com

Microchip
MPLAB X IDE; MPLAB XC32
TCP/IP Stack
www.microchip.com

AppConfig				
MyIPAddr		Val	b[0]	0x0101FEA9
w	w[0]	b[0]	0x0101	65193
w[1]	b[1]	b[2]	0x0101	257
v	v[0]	b[0]	0x0101	169
v[1]	b[1]	b[2]	0x0101	254
v[2]	b[3]	b[4]	0x0101	1
v[3]	b[5]	b[6]	0x0101	1

■ Screenshot 5. The IP address has to be manipulated a bit to make sense to us humans. It's a bit tricky to get 169.254.1.1 from 0x0101FEA9.

```
BYTE security = CPElements.securityType;
if (bssDescIsValid)
{
    switch (security)
    {
        case 0:
            TCPPutROMString(sktHTTP, (ROM
                BYTE*)"None");
            break;
        case 1:
            TCPPutROMString(sktHTTP, (ROM
                BYTE*)"WEP");
            break;
        case 2:
            TCPPutROMString(sktHTTP, (ROM
```

```
                BYTE*)"WEP");
            break;
        case 3:
            TCPPutROMString(sktHTTP, (ROM
                BYTE*)"WPA");
            break;
        case 4:
            TCPPutROMString(sktHTTP, (ROM
                BYTE*)"WPA");
            break;
        case 5:
            TCPPutROMString(sktHTTP, (ROM
                BYTE*)"WPA2");
            break;
        case 6:
            TCPPutROMString(sktHTTP, (ROM
                BYTE*)"WPA2");
            break;
        case 7:
            TCPPutROMString(sktHTTP, (ROM
                BYTE*)"WPA-AUTO");
            break;
        case 8:
            TCPPutROMString(sktHTTP, (ROM
                BYTE*)"WPA-AUTO");
            break;
        case 13:
            TCPPutROMString(sktHTTP, (ROM
                BYTE*)"WPA");
            break;
        default:
            // impossible to get here!
            break;
    }
}
```

Power of a PIC® MCU in your USB port...

Rapid USB Prototyping Stick and Single-chip C Compiler

Step-by-step guide to start C programming in minutes!

\$65 Special!

Inside view of Rapid USB

Capabilities & Features

- Pushbutton
- Text I/O to PC
- Two LEDs
- Easy Graphing of PIC® MCU Data

C COMPILER

For More Information, visit:
www.ccsinfo.com/NV114

CCS (262) 522-6500 x 35
sales@ccsinfo.com

Microchip Premier Partner

PIC® MCU is a registered trademark of Microchip Technology Inc.

f t y

Save Up To 60% On Electronic Components

Ethernet Mini Board
*Includes W5100 Embedded TCP/IP IC
*Works with Most Network Protocols
*3.3V or 5V Board Operation
*Ideal for Adding Network Connectivity to your Project

Only \$19.90

New Marine Solar Charge Controller
*Especially Designed for Marine Use
*Suitable for Both 12V and 24V Systems
*Waterproof and Corrosion Resistant

Only \$39.90

New microSD Mini Board
*Ideal for Saving and Recording Real-Time Readings
*Works with both 3.3V and 5V Microcontrollers
*Standard Communications Interface

Only \$9.40

We are your one-stop shop for Microcontroller Boards, PCB Manufacture and Electronic Components

www.futurlec.com

```

else
{
    TCPPutROMString(sktHTTP, (ROM
    BYTE*)"None");
}
}

```

We've come this far. So, let's put some detective work behind the `~wlan~` dynamic variable. We won't have to work very hard as **Screenshot 4** holds the key. `CPElements.networkType` holds a value of 0x02. A quick hop over to `WFApi.h` in the TCP/IP stack *Include* folder resolves the hexadecimal network type value:

```

typedef enum
{
    WF_INFRASTRUCTURE = 1,
    WF_ADHOC          = 2
} tWFNetworkType;

```

The song remains the same and so does the order from `HTTP2`. Here's the callback function that supports the `~wlan~` dynamic variable:

```

// =====
// = Sends the new/current WLAN setting type
// =====
void HTTPPrint_wlan(void)
{
    if(CPElements.networkType == WF_ADHOC)
        TCPPutROMString(sktHTTP, (ROM

```

```

        BYTE*)"AdHoc    (IBSS)");
    else
        TCPPutROMString(sktHTTP, (ROM
        BYTE*)"Infrastructure (BSS)");
}

```

The IP address data you see in **Screenshot 5** can be confusing. The `~ipaddr~` callback performs the translation from hexadecimal to human dot format:

```

void HTTPPrint_ipaddr(void)
{
    char ipAddr[16];
    sprintf(ipAddr,"%u.%u.%u.%u",
    AppConfig.MyIPAddr.v[0],
    AppConfig.MyIPAddr.v[1],AppConfig.
    MyIPAddr.v[2],AppConfig.MyIPAddr.v[3]);
    TCPPutString(sktHTTP, (BYTE *)ipAddr);
}

```

LED, Sensor, Motor, or Coffee Pot

If we can make an LED do our will with a web page, why not replace the LED with something more useful? Multiple devices can be monitored and controlled using the embedded web application framework we've discussed here. Throw out those chicken head knobs and add embedded web development to your design cycle.

NV



As low as...
\$9.95
each!

Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com



RF Specialists

Affordable RF Modules

Introducing the
LMX-ISM-242-SR and
LMX-ISM-242-LR



Data Loggers

Stand Alone and
Wireless Mesh Networking Logger



Industrial Bluetooth

OEM, Modules, Wireless Device Servers, RS-232
Long range options; low cost



ZigBee Pro

OEM Modules and USB ZigBee Sticks,
Mesh Networks



Low Power, High Performance Semiconductors

Excellent Sensitivity, Very Efficient Power Amplifiers

Narrow Band, General Purpose, TSSOP, RF Microcontrollers
Typical Sensitivities are in the range of -110 dBm to -126 dBm for 1.2 kbps

RF Design Services

Prepared to work with your in-house engineers,
or support your RF project from initial design to implementation.

Industrial • Military • Space • Medical • Smart Grid Metering • SCADA • Lighting Control



LEMOS
INTERNATIONAL

www.lemosint.com

866.345.3667

sales@lemosint.com

"Making your RF ideas into profitable products."

For complete product details, visit our webstore!!

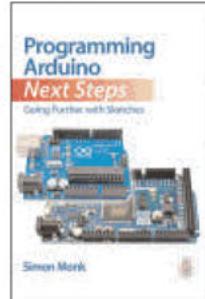
The Nuts & Volts WEBSTORE

GREAT FOR DIYers!

Programming Arduino Next Steps: Going Further with Sketches by Simon Monk

Arduino guru Simon Monk reveals advanced programming techniques for Arduino! *Programming Arduino Next Steps: Going Further with Sketches* is the must-have follow-up to Monk's bestseller, *Programming Arduino: Getting Started with Sketches*. Aimed at experienced programmers and hobbyists who have mastered the basics, this book takes you "under the hood" of the Arduino, revealing professional-level programming secrets.

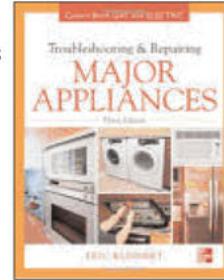
\$20.00



Troubleshooting and Repairing Major Appliances 3rd Edition by Eric Kleinert

Fully updated for current technologies and packed with hundreds of photos and diagrams, this do-it-yourself guide shows you how to safely install, operate, maintain, and fix gas and electric appliances of all types. *Troubleshooting and Repairing Major Appliances, Third Edition* provides easy-to-follow procedures for using test meters, replacing parts, reading circuit diagrams, interpreting fault and error codes, and diagnosing problems.

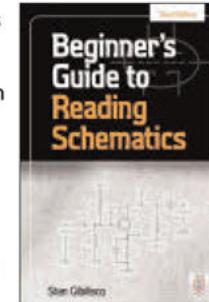
\$60.95



Beginner's Guide to Reading Schematics, 3E by Stan Gibilisco

Navigate the roadmaps of simple electronic circuits and complex systems with help from an experienced engineer. With all-new art and demo circuits you can build, this hands-on, illustrated guide explains how to understand and create high-precision electronics diagrams. Find out how to identify parts and connections, decipher element ratings, and apply diagram-based information in your own projects.

\$25.00



30 Arduino Projects for the Evil Genius: Second Edition by Simon Monk

Fully updated throughout, this do-it-yourself guide shows you how to program and build fascinating projects with the Arduino Uno and Leonardo boards, and the Arduino 1.0 development environment. *30 Arduino Projects for the Evil Genius, Second Edition*, gets you started right away with the simplified C programming you need to know, and demonstrates how to take advantage of the latest Arduino capabilities.

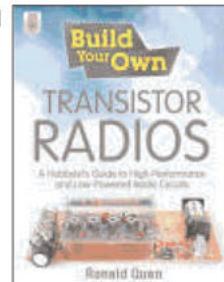
\$25.00



Build Your Own Transistor Radios by Ronald Quan A Hobbyist's Guide to High Performance and Low-Powered Radio Circuits

Create sophisticated transistor radios that are inexpensive yet highly efficient. Inside this book, it offers complete projects with detailed schematics and insights on how the radios were designed. Learn how to choose components, construct the different types of radios, and troubleshoot your work.

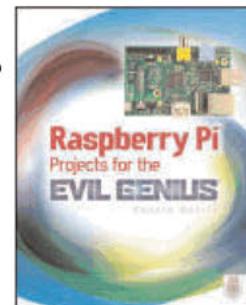
*Paperback, 496 pages
\$49.95



Raspberry Pi Projects for the Evil Genius by Donald Norris

This wickedly inventive guide shows you how to create all kinds of entertaining and practical projects with the Raspberry Pi operating system and programming environment. Each fun, inexpensive Evil Genius project includes a detailed list of materials, sources for parts, schematics, and lots of clear, well-illustrated instructions for easy assembly. The larger workbook-style layout makes following the step-by-step instructions a breeze.

\$25.00



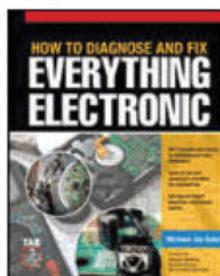
How to Diagnose and Fix Everything Electronic by Michael Jay Geier

Master the Art of Electronics Repair

In this hands-on guide, a lifelong electronics repair guru shares his tested techniques and invaluable insights. *How to Diagnose and Fix Everything*

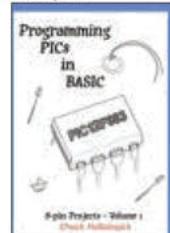
Electronic shows you how to repair and extend the life of all kinds of solid-state devices, from modern digital gadgetry to cherished analog products of yesteryear.

\$24.95



Programming PICs in Basic by Chuck Hellebuyck

If you wanted to learn how to program microcontrollers, then you've found the right book! Microchip PIC microcontrollers are being designed into electronics throughout the world and none is more popular than the eight-pin version. Now the home hobbyist can create projects with these little microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software language. Chuck Hellebuyck introduces how to use this development setup to build useful projects with an eight-pin PIC12F683 microcontroller. **\$14.95**



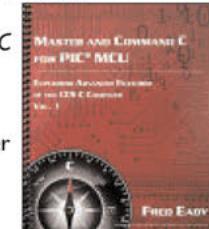
Master and Command C for PIC MCUs

by Fred Eady
Master and Command C for PIC MCU, Volume I aims to help readers get the most out of the Custom Computer Services C compiler for PIC

microcontrollers.

The author describes some basic compiler operations that will help programmers particularly those new to the craft create solid code that lends itself to easy debugging and testing. As Eady notes in his preface, a single built-in CCS compiler call (`output_bit`) can serve as a basic aid to let programmers know about the "health" of their PIC code.

\$14.95



Order online @ www.store.nutsvolts.com
Or CALL 1-800-783-4624 today!

BOOK & KIT COMBOS

16-Bit Complete Combo



Combo Price \$319.95

For complete details visit our webstore @ www.nutsvolts.com

Beginner's Guide Complete Combo!



Combo Price
\$229.95

For complete details, visit our webstore @ www.nutsvolts.com.

Arduino Handheld Prototyper System

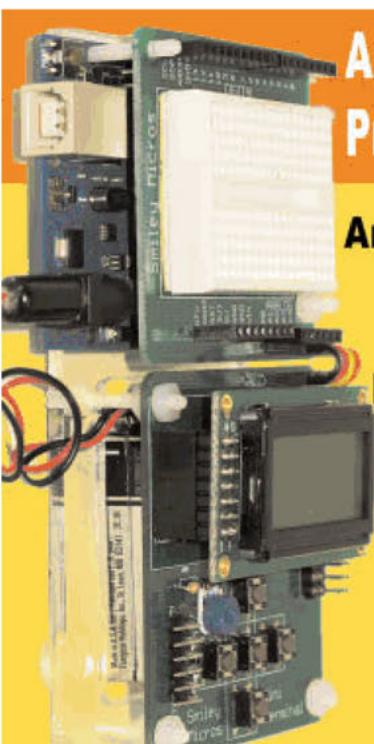
Arduino Proto Shield
\$16.99

I2C Mini Terminal
\$29.99

Handheld Base
\$13.99

All three*: \$59.99

From Smiley's Workshop

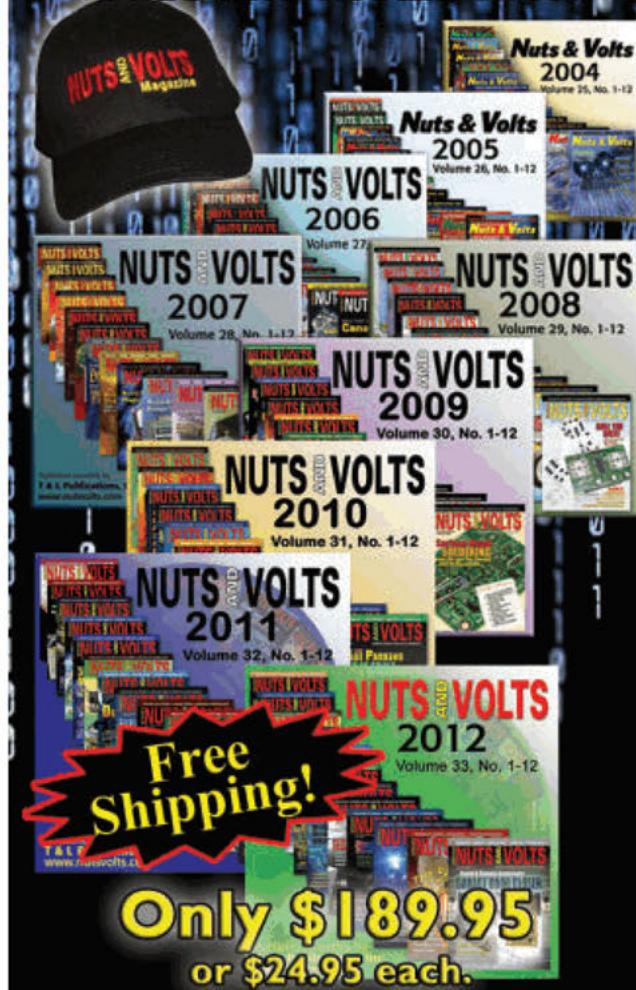


*Arduino and battery not included

CD-ROM SPECIAL

Nuts & Volts 9 CD-ROMs & Hat Special!

That's 108 issues.
Complete with supporting
code and media files.



Free
Shipping!
Only \$189.95
or \$24.95 each.



The Nuts & Volts Pocket Ref

All the info you need at your fingertips!

This great little book is a concise
all-purpose reference featuring
hundreds of tables, maps,
formulas, constants &
conversions.

Only \$12.95

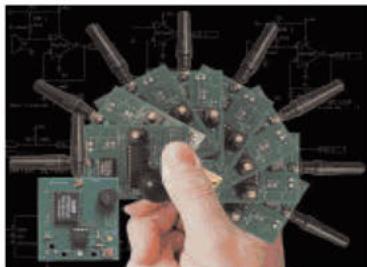
AND it still fits in
your shirt pocket!

Visit <http://store.nutsvolts.com> or call (800) 783-4624

**CALL 1-800-783-4624 today!
Order online @ www.nutsvolts.com**

PROJECTS

Super Detector Circuit Set



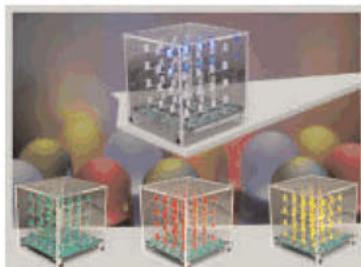
Pick a circuit!

With one PCB you have the option of detecting wirelessly: temperature, vibration, light, sound, motion, normally open switch, normally closed switch, any varying resistor input, voltage input, mA input, and tilt, just to name a few.

Subscriber's Price \$32.95

Non-Subscriber's Price \$35.95

3D LED Cube Kit



This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes.

Colors available: Green, Red, Yellow & Blue

Subscriber's Price \$57.95

Non-Subscriber's Price \$59.95

Radiation Monitor Alarm Kit



This is an inexpensive surface-mount project that is good for beginners to start with. This kit has its own printed circuit board (PCB) which makes mounting the components easy. Plus, it comes in a pocket size shielded aluminum case measuring 3.5" in length and 1" in diameter. The on-off switch is a pushbutton on the bottom.

Subscriber's Price \$32.45

Non-Subscriber's Price \$33.95

Geiger Counter Kit

As seen in the March 2013 issue.

This kit is a great project for high school and university

students. The unit detects and displays levels of radiation, and can detect and display dosage levels as low as one micro-roentgen/hr. The LND712 tube in our kit is capable of measuring alpha, beta, and gamma particles.

Partial kits also available.

Subscriber's Price \$145.95

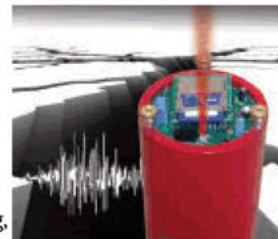
Non-Subscriber's Price \$148.95



Seismograph Kit

As seen in the May 2012 issue.

Now you can record your own shaking, rattling, and rolling.



The Poor Man's Seismograph is a great project /device to record any movement in an area where you normally shouldn't have any. The kit includes everything needed to build the seismograph. All you need is your PC, SD card, and to download the free software to view the seismic event graph.

Subscriber's Price \$79.95

Non-Subscriber's Price \$84.95

Battery Marvel

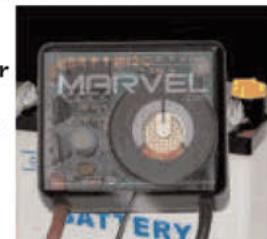
As seen in the November 2011 issue.

Battery Marvel helps protect cars, trucks, motorcycles, boats, and any other 12V vehicles from sudden battery failure. This easy-to-build kit features a single LED that glows green, yellow, or red, indicating battery health at a glance. An extra-loud piezo driver alerts you to any problems.

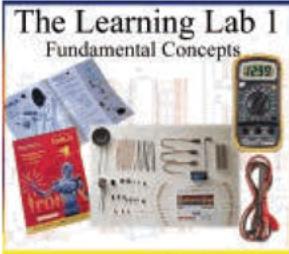
Details, please visit our website.

Subscriber's Price \$18.95

Non-Subscriber's Price \$19.95



FOR BEGINNER GEEKS!

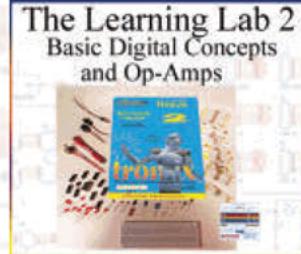


\$59.95

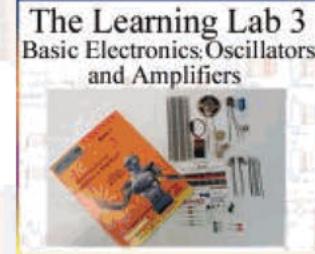
The labs in this series — from GSS Tech Ed — show simple and interesting experiments and lessons, all done on a solderless circuit board.

As you do each experiment, you learn how basic components work in a circuit, and continue to build your arsenal of knowledge with each successive experiment.

For more info and a promotional video, please visit our webstore.



\$49.95



\$39.95

CLASSIFIEDS

COMPONENTS



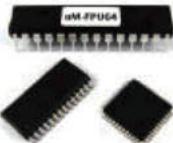
RF PARTS™ CO.

Tubes, Transistors, Power Components
Email: rfp@rfparts.com • Web: www.rfparts.com

800-737-2787 Fax 888-744-1943

uM-FPU64

64-bit Floating Point Coprocessor



64-bit and 32-bit IEEE 754 compatible

SPI or I2C interface, 3.3V Supply, 5V Tolerant I/O

Extensive floating point support, FFT, matrix operations

GPS input, local device support, Flash functions, IDE

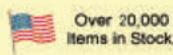
Robotics
Navigation
Sensor Modules
Embedded Systems

DIP-28, SOIC-28,TQFP-44
www.micromegacorp.com

Additional products...
uM-FPU V3.1 32-bit FPU
uM-PWM1 Servo Coprocessor

SURPLUS

SURPLUS ELECTRONIC PARTS & ACCESSORIES



Over 20,000
items in Stock.

Cables
Connectors
Displays
Fans

Hardware
LEDs
Motors
Potentiometers

Relays
Semiconductors
Service Manuals
Speakers

Switches
Test Equipment
Tools
VCR Parts

Surplus Material Components
SMC ELECTRONICS
www.smcelectronics.com

No Minimum Order.
Credit Cards and PAYPAL Accepted.
Flat \$4.95 per order USA Shipping.

LIGHTING



www.ReactiveLighting.com

SECURITY

SPY EQUIPMENT

BEST PRICES GUARANTEED
on CCTV and counter spy
products!

914.699.2294

www.mjelectronics.com

WANT TO BUY



Excess Solutions
156 S. Milpitas Blvd.
Milpitas, CA 95035

Come in and see our retail store!

We Buy & Sell Excess
Electronic Inventory

(408) 262-3900

www.excesssolutions.com

HARDWARE WANTED

DEC EQUIPMENT WANTED!!!

Digital Equipment Corp.
and compatibles.
Buy - Sell - Trade

CALL KEYWAYS 937-847-2300
or email buyer@keyways.com

ENERGY SOLUTIONS



Save Money On
Your Electric Bill
www.sunny-solutions-shop.com

800-315-0309

ROBOTICS

Ultrasonic Distance Sensing Made EZ

LV-MaxSonar®-EZ™

- Most popular MaxSonar
- Multiple interfaces
- Easy integration
- 1 inch resolution
- Calibrated beam patterns
- Starting at \$29.95



HRXL-MaxSonar®-WR™

- High noise tolerance
- IP67 rated
- 1 mm resolution
- Multi-Sensor operation
- Calibrated beam pattern
- Starting at \$109.95



XL-MaxSonar®-EZ™

- Great for UAV's and robotics
- Incredible noise immunity
- Small in size
- 1cm resolution
- Automatic calibration
- Starting at \$39.95



www.maxbotix.com

CONTROLLERS

INTEGRATED Ethernet PLCs for OEMs



From \$119

NANO-10
- ETHERNET / Modbus TCP/IP
- 8 DI/Os, 2 AI/Os
- RS485
- PWM / PID / Stepper Control
- Ladder + BASIC Programming

tel : 1 877 TRI-PLCS
web : www.triplc.com/nv.htm



TRI
TRIANGLE
RESEARCH
INTERNATIONAL

SERVICES

MIND CONTROL ELECTRONIC ATTACKS CUSTOM DESIGNS

Also, Energy, Security, Survival
Biomed, RF, Audio, Ultrasound
Video, Phones, Patents, More ...
www.lonestarconsultinginc.com

KITS/PLANS

QKITS.COM

We have lots of fun
electronic projects
for everyone of any
skill level!

Stock up now for your
winter building!

FREE SHIPPING

On all ground orders over \$175

www.qkits.com

1-888-GO-4-KITS



MAGNETIC LEVITATION KIT

Float small objects
in thin air.

From \$39.75

www.arttec.net/Levitation

D.I.Y. LASER TAG

www.LaserTagParts.com

PARTS | KITS | SCHEMATICS | FIRMWARE
FORUM | STORE | HOBBY & COMMERCIAL

MISC FOR SALE

HUGE CLOSE-OUT SALE!

All kinds of electronic
products and parts!

Test equipment, schematics, tubes,
audio-video equipment ... too
numerous to list. Call for availability.

loustvelectronics@verizon.net

215.271.1138 or 215.468.3028

215.432.0333

Call Anytime/24-7

HARDWARE

Single and Dual Axis Light Seeking Solar Tracker Controls

www.theanalogguy.com

BOOKS

Free AI Book and Software.

IBM PC. Write:

davethecoyote@hotmail.com

Nuts
& Volts
Mobile
App!



Check it out now at
your favorite
App Store!

>>> QUESTIONS

Amplifier Hum

I picked up a "hum blocker" for my guitar amp, only to discover that all it does is disconnect the ground from a three-prong outlet.

It seems to work. The hum — whether from a ground loop or pickup from fluorescent lights — is much less noticeable. My question: Is this device safe to use?

#1141

Steven Sewal
via email

Which Way Should I Go?

I'm new to electronics, recently retired, and in need of some direction. Should I spend my time learning about resistors, capacitors, and transistors, or start with an Arduino or other microcontroller?

#1142

Dan Johnson
Oceanside, CA

Prototype to Production

After about five years of building prototypes, I've finished the design for a device that I believe can be sold to the masses. What's my next step? Is it worth spending money on a patent attorney? Is there a clearing house of sorts that connects developers in the US with production houses in China? Has anyone been down this road and care to share their experience?

#1143

Jeremy Williams
via email

Smartphone Compass

My smartphone has a built-in compass that seems to be unaffected

by local metal structures or magnets. Is the phone using cell tower triangulation or some other method to determine direction? If so, does the phone indicate true north or magnetic north?

#1144

Robin Stahl
Cincinnati, OH

Schematics Needed

I found an old transistor radio in my parent's attic. It's one of the first transistor radios made by RCA. Is there a good source for schematics on old radios and other electronics? I've tried the usual Google searches, but turned up nothing.

#1145

Bruce Tinney
Tuscaloosa, AL

Telescope Position Control

I'm looking for a motor drive for my 6" refractor telescope. A stepper motor should give me position control, but big steppers are expensive. DC motors are cheaper, but require a complex gear box and a sensor to determine position. Is there an affordable option that provides me with the positioning benefits of a stepper? My telescope and camera back weigh about 6 lbs total.

#1146

Alex Forsyth
Columbus, OH

>>> ANSWERS

[#10133 - October 2013] PWM Power and Short-circuit Protection

I'm trying to design a PWM circuit to control power to a resistive heater

Send all questions and answers by email to forum@nutsvolts.com or via the online form at www.nutsvolts.com/tech-forum

All questions AND answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. All

submissions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted by readers and **NO GUARANTEES**

strip for my camera/telescope. Recently, my nice commercial unit let the magic smoke out after a short on the output side, so I'm not eager to replace it with another of the same make.

The problem is I'm a complete newbie. I'm using a 4093 for the PWM part of the circuit and a pMOSFET (IRF9510) for the power switching. If I didn't want the short-circuit protection, I would be done. I've found a few schematics online but am not sure how to integrate them into my circuit.

Here's the schematic (**Figure 1**). I've put together, however, there are two things that bother me. First, I've essentially glued the short-circuit protection onto the output of the pMOSFET which means I have another diode drop in the output. That seems like it should be unnecessary and it seems like I ought to be able to put the IRF9510 where the SK100 is, but I'm not sure how to do that correctly.

Second, the whole thing will be powered off a marine battery and I'd like some input protection/isolation of the control part to avoid the possibility of frying the 4093 from transients when hooking up power.

Any help is greatly appreciated!

I made some mods to your circuit in **Figure 1**. With this circuit, Q1 will go directly to the load. With the load there, Q2 will turn on; its collector will go low. The output of U1-D will be high, so U1-B and U1-C will invert the other input. If the load is shorted, Q2 will be off and the input to U1-D will

WHATSOEVER are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals.

Always use common sense and good judgment!

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

be high; its output will be low. So, the inputs to U1-B and U1-C will be low. This will force the outputs high, and turns off Q1.

[#11131 - November 2013]

Tesla Coil Theory

How does a *Tesla coil* actually work? I'm especially interested in the relationship between the primary winding and secondary winding that creates the spark.

#1 The Tesla coil uses a resonant transformer with tuned primary and secondary coils to produce high frequency current at very high voltage. A resonant transformer is like a child's swing, or pendulum; by pushing repeatedly at just the right time on each cycle, the pendulum can be made to swing much farther than from any one push. Also, as in the case of a pendulum, the trick is to give a short sharp push and to let go, which may be done in a Tesla coil using a spark gap that is conductive for only a part of the cycle. When the circuit is broken, the high dv/dt (rate of change of voltage) in the inductance of the primary circuit creates a high voltage wave. In addition, like any transformer, the Tesla coil multiplies the output voltage by the turns ratio of the primary to the secondary coil.

A simple Tesla coil could be made from an electromagnetic door buzzer (**Figure 2**).

Though the battery might be just 1.5 volts, a 60 volt neon lamp could be lit by connecting it to the contact and

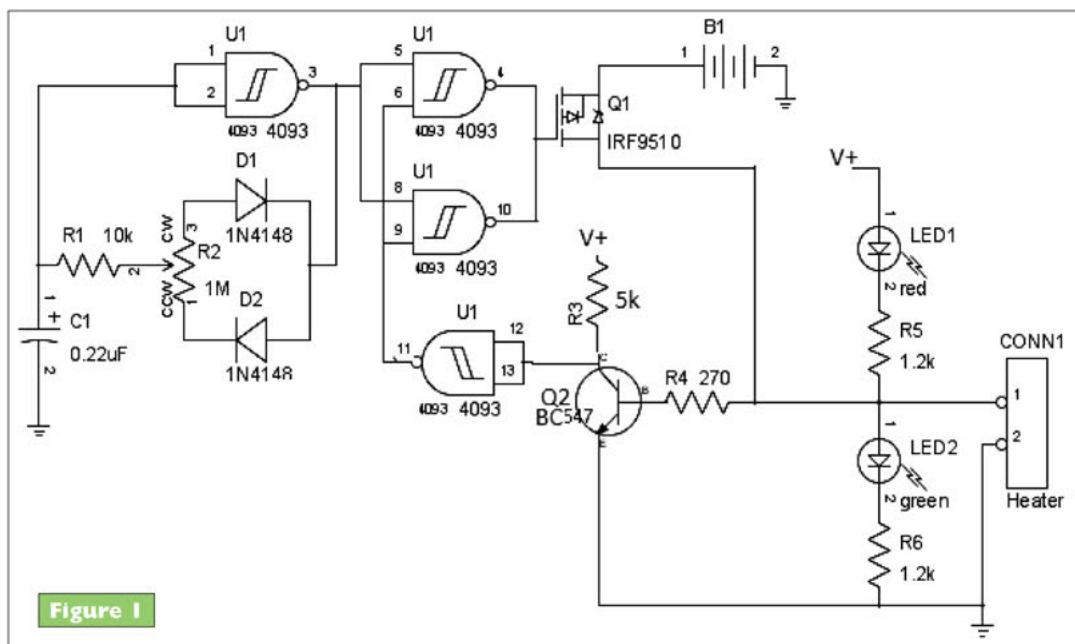


Figure 1

Introductory offer! Save \$10 with this code: helloNV

Introducing the **Gobo!** Stereo Audio Amp Kit



- Complete start to finish kit! Includes all the parts you need to build it.
- Custom analog-based design produces warm, clean, retro sound.
- Easy-to-follow, step-by-step instructions, rich with quality photos. Videos too!
- Designed and documented so you can learn and modify as you grow.
- Affordable without compromising quality. High-end, audio-grade components.

boxed
KIT AMPS

Get your Gobo at www.boxedkitamps.com today!

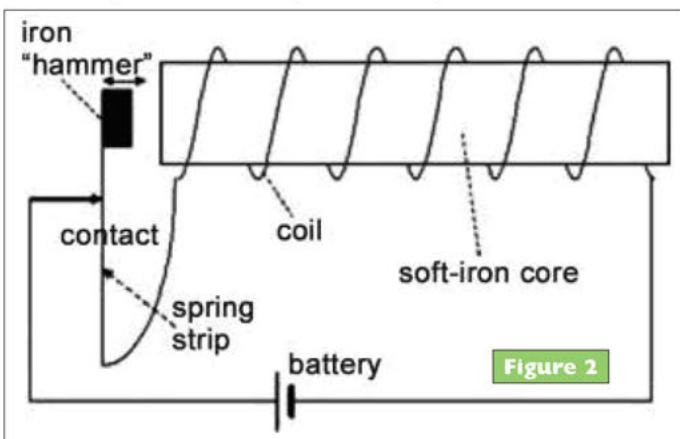


Figure 2

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum

spring strip. Here, the high voltage is due to the high dv/dt as the moving spring strip suddenly opens the circuit. Winding a secondary coil with many more turns over the existing primary coil would increase the output voltage on the secondary by the turns ratio. Placing the right value of capacitor across the primary coil to tune it to the self-resonant frequency of the secondary would increase the voltage even more.

See http://en.wikipedia.org/wiki/Tesla_coil and www.hvtesla.com/tuning.html for more information.

Bart Bresnik
via email

#2 Go to www.teslacoidesign.com by Kevin Wilson. It is very well done and has a nice and brief explanation of the theory of operation and how to build it safely.

Phil
Mt Airy, MD

[#11133 - November 2013] Filter Caps and Power Supplies

This has to do with electrolytic aluminum filter caps for switching power supplies.

No matter what type filter cap I try, they blow out (become pregnant) after months or a few years. I repaired cable boxes for many years that had the exact same problem.

This is only a three volt supply at about two amps. Ten volt 1,000 mfd caps are used in the stock supply. Also, a three amp Schottky diode (burns up) supplies the DC to a 15 amp logic N-channel MOSFET with a heatsink. It gets hot. Then, the output of it gets a cap, a choke, and a cap. Nicely filtered three volts.

This is my final change-out and it is lasting the longest. So far, no blow outs, but it has only been seven months.

Now, the two five amp Schottky diodes in parallel. Using only one still gets super hot. Caps 25 volts at 1,000 mfd. I'm only using general type filter

caps at 20% 105° C. Why has this been such a big problem?

The cap that usually blows is the first one after the MOSFET. I see no spikes on the output of the MOSFET either. I could use a TO-220 pack with dual diodes in it, but no room. The two five amps in parallel work just great and only get warm.

#1 The quick answer to your inquiry is that you're using the wrong type of filter capacitors.

From the description given of the power supply — three volts, two amperes (six watts), and the heating problems encountered — I surmise that you have a small flyback switching power supply that is not operating very efficiently. If the supply is rated for a mains input range of 90-130 volts AC and you're operating near the top end of that range — 120 volts or so — the ON time of the switch transistor will be quite short relative to the OFF time. All of the input power to the flyback transformer must be delivered during that short ON time, so the switch current will be high. Similarly, the flyback (secondary) current pulse through the output diode will be high because it must deliver all of its energy to the output capacitor in a short time. Both of these conditions serve to elevate the operating temperature of the switch transistor and output diode.

Finally, that output current pulse is dumped into the output capacitor. A real capacitor can be visualized as an ideal capacitor in series with a small resistance; the latter is known as the Equivalent Series Resistance (ESR). You need to use capacitors having a very low ESR value and a high ripple-current rating. I would expect that your output capacitors are seeing very high instantaneous ripple currents. Capacitor heating is a function of the ESR value and of the square of the RMS value of the ripple current. A suitable capacitor might be a Panasonic EEU-FR1E102, available from Digi-Key (part number P14424-

ND, \$0.91 each). The ESR of this device is 0.020 ohms and it will tolerate over two amperes RMS.

As far as paralleling diodes goes, I've had bad experience with that. You cannot guarantee exactly when the diode will switch from non-conduction to conduction, so for a very short instant one diode may be exposed to the full current pulse. I can't visualize why you have room to fit two five ampere Schottky diodes but not room for one 10 ampere TO-220 package.

The reason that the capacitor nearest the MOSFET switch is the first to be destroyed probably relates to the board layout, and the fact that the other capacitors have additional lead inductance (including the etched conductors) in series with them. If possible, try to equalize distribution of current from the switch to each of the capacitors, and do the same for their returns to the common bus.

I hope these suggestions help.

Peter A. Goodwin
Rockport, MA

#2 The likely cause is the high frequency ripple; aluminum electrolytic capacitors don't deal well with high frequencies (high loss factor) or a large AC component of a waveform (capacitor may be depolarized).

You could substitute a tantalum capacitor, and/or parallel some ceramic capacitors across the aluminum one, e.g. 0.01 µF and 0.5 µF (or 10 nF and 500 nF, if you prefer), keeping the leads of the ceramic capacitors short. You could also use a small ferrite choke in series with the capacitor to decrease the AC component of the waveform.

A schematic of the power supply would help to pinpoint the problem.

Bart Bresnik
via email

Write for Nuts & Volts!
Request a copy of our
writer's guidelines at
editor@nutsvolts.com

case. There's also the safety issue of using neutral as a false ground. Given my experience with holiday lighting – which typically includes the use of at least one questionable extension cord – I'd opt for an inexpensive plastic enclosure and keep ground out of the equation.

Bryan Bergeron, Editor

(Un)Zip IT

I enjoyed the PICAXE Primer column on Python and RPi. I am using Python more (have done a fair amount of C) and found myself moving a Python module to the RPi

this past Saturday. This Sunday evening, I was looking at the column in the October 2013 issue of *Nuts & Volts*. In the article, there was a mention that it was difficult to unzip a file on the Raspberry Pi.

I move info to the RPi from my laptop via network utilization, but that's not relevant to the next point which is how to unzip. I used the 'gunzip' command, after first renaming my file's zip extension to .gz. File: myfile.zip renamed to myfile.gz.

On the RPi, the command is *mv myfile.zip myfile.gz*. The 'zip' file is no longer present in the directory. The command is then *gunzip myfile.gz*.

The unzipped file is now in your directory (and the zipped file

is no more).

Caveat 1: If you want to save a copy of the myfile.zip or myfile.gz, you can create a copy (with a different name) before doing any of the above commands.

The command *cp myfile.zip myfile_copy1.zip* creates a copy of the original file under a different name. If you don't want to have a renamed file, you can save a copy of the file under its original name in another directory.

Possible Caveat 2: I am using the Occidentalis distribution of the RPi OS – this is from Adafruit. I think that gunzip would be available on the standard distribution, but I can't directly confirm that.

Margaret Lyell

■ LOOK FOR ■ SEARCH FOR ■ FIND
Find your favorite advertisers here!

AMATEUR RADIO AND TV

Ramsey Electronics 82-83

BATTERIES/CHARGERS

Cunard Associates 27

BUYING ELECTRONIC SURPLUS

Weirdstuff Warehouse 27

CCD CAMERAS/VIDEO

Ramsey Electronics 82-83

CIRCUIT BOARDS

AP Circuits 73

Cunard Associates 27

Dimension Engineering 51

ExpressPCB 18

Front Panel Express LLC 22

Futurlec 72

GHI Electronics 16

PCB Pool 22

Reach Technology 27

Saelig Company Inc. 61

CNC ROUTER

Next Wave Automation 6

Probotix 50

COMPONENTS

Electronix Express 11

Futurlec 72

Jameco Back Cover

COMPUTER

Hardware

Technologic Systems 17

Weirdstuff Warehouse 27

Microcontrollers / I/O Boards

Abacom Technologies 23

Custom Computer Services 72

GHI Electronics 16

Images Co. 43

microEngineering Labs 42

MikroElektronika 3

Pololu Robotics & Electronics 19

Technologic Systems 17

Software

Technologic Systems 17

DESIGN/ENGINEERING/REPAIR SERVICES

ExpressPCB 18

Front Panel Express LLC 22

Images Co. 43

PCB Pool 22

Technologic Systems 17

EDUCATION

Boxed Kit Amps 79

Command Productions 7

Next Wave Automation 6

NKC Electronics 27

Oscium 5

Poscope 33

EMBEDDED TOOLS

NetBurner 2

ENCLOSURES

Front Panel Express LLC 22

ETHERNET

Custom Computer Services 72

EVENTS

International CES® 66

Tropical Hamboree/Mini

Maker Faire 51

HI-FI AUDIO

Boxed Kit Amps 79

KITS & PLANS

Boxed Kit Amps 79

Custom Computer Services 72

NetBurner 2

NKC Electronics 27

QKITS 27

Ramsey Electronics 82-83

Reach Technology 27

MISC./SURPLUS

All Electronics Corp. 37

Front Panel Express LLC 22

Weirdstuff Warehouse 27

MOBILE RADIO MOUNTS

E Tip Inc. 27

MOTORS

Jameco Back Cover

PROGRAMMERS

Futurlec 72

microEngineering Labs 42

MikroElektronika 3

RF TRANSMITTERS/RECEIVERS

Abacom Technologies 23

E Tip Inc. 27

ROBOTICS

GHI Electronics 16

Jameco Back Cover

Lemos International Co. 73

Oscium 5

Pololu Robotics & Electronics 19

Probotix 50

Robot Power 27

SATELLITE

Lemos International Co. 73

SECURITY

E Tip Inc. 27

TEST EQUIPMENT

Dimension Engineering 51

Electronic Design Specialists 50

Images Co. 43

NKC Electronics 27

Oscium 5

Poscope 33

Saelig Company Inc. 61

TOOLS

MikroElektronika 3

NetBurner 2

Next Wave Automation 6

PanaVise 23

Patco 43

PCB Pool 22

Pololu Robotics & Electronics 19

Poscope 33

Probotix 50

QKITS 27

Ramsey Electronics 82-83

Reach Technology 27

TOUCH SCREEN DISPLAY MODULES

Reach Technology 27

WIRE PROCESSING/STRIPPING

Patco 43

ADvertiser INDEX

Abacom Technologies 23

All Electronics Corp. 37

AP Circuits 73

Boxed Kit Amps 79

Command Productions 7

Cunard Associates 27

Custom Computer Services 72

Dimension Engineering 51

Electronic Design Specialists 50

Electronix Express 11

Futurlec 72

GHI Electronics 16

Images Co. 43

International CES® 66

Jameco Back Cover

Lemos International Co. 73

microEngineering Labs 42

MikroElektronika 3

NetBurner 2

Next Wave Automation 6

NKC Electronics 27

Oscium 5

PanaVise 23

Patco 43

PCB Pool 22

Pololu Robotics & Electronics 19

Poscope 33

Probotix 50

QKITS 27

Ramsey Electronics 82-83

Reach Technology 27

Robot Power 27

Saelig Company Inc. 61

Technologic Systems 17

Tropical Hamboree/Mini

Maker Faire 51

Weirdstuff Warehouse 27



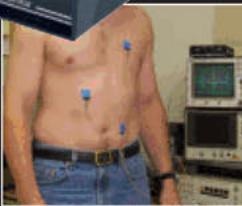
Electrocardiogram ECG Heart Monitor

- ✓ Visible and audible display of your heart rhythm!
- ✓ Bright LED "Beat" indicator for easy viewing!
- ✓ Re-usable hospital grade sensors included!
- ✓ Monitor output for professional scope display
- ✓ Simple and safe 9V battery operation



When we think of a New Year, we often start with our proverbial list of New Year's resolutions. Often starting with weight loss goals as a result of all the December holiday parties, (including the Buffalo Wings of course!) they typically also include a new attitude for a healthier life style. While we are frequently reminded that February is national Heart Smart month, we think the first of the Year is no better time to start! Every month should be Heart Smart month. Heart Smart is a way of life, and certainly shouldn't be limited to one month a year. We kept that in mind when we designed the ECG1!

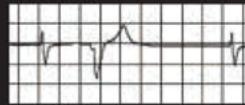
Not only will building an actual ECG be a thrill, but you'll get hands-on knowledge of the relationship between electrical activity and the human body. Each time the human heart beats, the heart muscle causes small electrical changes across your skin. By monitoring and amplifying these changes, the ECG1C detects the heartbeat and allows you to accurately display it, and hear it, giving you a window into the inner workings of the human heart and body!



One of our engineers/guinea pigs, checking his heart!

Use the ECG1C to astound your physician with your knowledge of ECG/EKG systems. Enjoy learning about the inner workings of the heart while, at the same time, covering the stage-by-stage electronic circuit theory used in the kit to monitor it. The three probe wire pick-ups allow for easy application and experimentation without the cumbersome harness normally associated with ECG monitors.

The documentation with the ECG1C covers everything from the circuit description of the kit to the circuit description of the heart! Multiple "beat" indicators include a bright front panel LED that flashes with the actions of the heart along with an adjustable level audio speaker output that supports both mono and stereo hook-ups. In addition, a monitor output is provided to connect to any standard oscilloscope to view the traditional style ECG/EKG waveforms just like you see in a real ER or on one of the medical TV shows!



On a personal note... See the display to the left? That's me! In between writing this mostly ad copy, catalog copy, and plethora of other tasks here, I noticed some skipped beats in my pulse! An immediate cardiac check found I had something called Trigeminus, or PVCs that occur at intervals of 2 normal beats to one PVC! And I saw it with our ECG1 kit!

Look what I found!

The fully adjustable gain control on the front panel allows the user to custom tune the differential signal picked up by the probes giving you a perfect reading and display every time! 10 hospital grade re-usable probe patches are included together with the matching custom case set shown. Additional patches are available in 10-packs. Operates on a standard 9VDC battery (not included) for safe and simple operation. Note, while the ECG1C professionally monitors and displays your heart rhythms and functions, it is intended for hobbyist usage only. If you experience any cardiac symptoms, seek proper medical help immediately!

ECG1C	Electrocardiogram Heart Monitor Kit With Case & Patches	\$44.95
ECG1WT	Electrocardiogram Heart Monitor, Factory Assembled & Tested	\$89.95
ECGP10	Electrocardiogram Re-Usable Probe Patches, 10-Pack	\$4.95

Digital Voice Changer

This voice changer kit is a riot! Just like the expensive units you hear the DJ's use, it changes your voice with a multitude of effects! You can sound just like a robot, you can even add vibrato to your voice! 1.5W speaker output plus a line level output! Runs on a standard 9V battery.



MK171 Voice Changer Kit

\$14.95

Steam Engine & Whistle

Simulates the sound of a vintage steam engine locomotive and whistle! Also provides variable "engine speed" as well as volume, and at the touch of a button the steam whistle blows! Includes speaker. Runs on a standard 9V battery.



MK134 Steam Engine & Whistle Kit

\$11.95

5A PWM Motor Controller

This handy controller uses a pulse width modulated output to control the speed of a motor without sacrificing torque!



Handles a continuous current of 5A and includes an LED to indicate speed, as well as an oversized gold heatsink! Also available factory assembled.

CK1102 5A PWM Motor Controller Kit

\$14.95

Audio Recorder & Player

Record and playback up to 8 minutes of messages from this little board! Built-in condenser mic plus line input, line & speaker outputs. Adjustable sample rate for recording quality. 4-switch operation that can be remote controlled! Runs on 9-12VDC at 500mA.



K8094 Audio Recorder/Player Kit

\$32.95

Laser Trip Sensor Alarm

True laser protects over 500 yards! At last within the reach of the hobbyist, this neat kit uses a standard laser pointer (included) to provide both audible and visual alert of a broken path. 5A relay makes it simple to interface! Breakaway board to separate sections.

LTS1 Laser Trip Sensor Alarm Kit

\$29.95

Electronic Watch Dog

A barking dog on a PC board! And you don't have to feed it! Generates 2 different selectable barking dog sounds. Plus a built-in mic senses noise and can be set to bark when it hears it! Adjustable sensitivity! Unlike my dog, eats 2-8VAC or 9-12VDC, it's not fussy!



K2655 Electronic Watch Dog Kit

\$39.95

Where Electronics...
Is Always Fun!

- ✓ Build It!
- ✓ Learn It!
- ✓ Achieve It!
- ✓ Enjoy It!

The Learning Center!

PL130A

SM200K

AMFM108K

SP1A

AM2

AMF7

SR3

AK870

PL200

PL300

Beginners To Advanced... It's Fun!

- ✓ Learn, build, and enjoy!
- ✓ 130, 200, 300, & 500 in one electronic labs!
- ✓ Practical through hole and SMT soldering labs!
- ✓ Integrated circuit AM/FM radio lab!
- ✓ AM, AM/FM, and SWL radio labs!
- ✓ Radio Controlled (RC) car!
- ✓ Beginner's non-soldering kits!

For over 4 decades we've become famous for making electronics fun, while at the same time making it a great learning experience. As technology has changed over these years, we have continued that goal!

PL130A Gives you 130 different electronic projects together with a comprehensive learning manual describing the theory behind all the projects.

PL200 Includes 200 very creative fun projects and includes a neat interactive front panel with 2 controls, speaker, LED display and a meter.

PL300 Jump up to 300 separate projects that start walking you through the learning phase of digital electronics.

PL500 The ultimate electronics lab that includes 500 separate projects that cover it all, from the basics all the way to digital programming.

SP1A Whether young or old, there's always a need to hone your soldering skills. Either learn from scratch or consider it a refresher, and end up with a neat little project when you're done!

SM200K Move up to Surface Mount Technology (SMT) soldering, and learn exactly how to solder those tiny little components to a board!

AMFM108K We not only take you through AM and FM radio theory but we guide you through IC's. When you're done you've built yourself an IC based AM/FM radio that works great!

AM2 Learn the complete theory of AM broadcast radio and end up with a highly sensitive AM radio receiver!

AMF7 Step up to AM/FM with this multi-band lab and learn the basics of both bands. The FM tuner is factory assembled and aligned!

SR3 Enter the world of SWL with this short-wave radio learning lab covering 6-8MHz and 12-18MHz!

AK870 One of the most exciting electronic learning kits that the kids will love! Build a complete RC speedster from the ground up! 7 remote functions!

PL130A	130-In-One Lab Kit	\$39.95
--------	--------------------	---------

PL200	200-In-One Lab Kit	\$84.95
-------	--------------------	---------

PL300	300-In-One Lab Kit	\$109.95
-------	--------------------	----------

PL500	500-In-One Lab Kit	\$249.95
-------	--------------------	----------

SP1A	Through Hole Soldering Lab	\$9.95
------	----------------------------	--------

SM200K	SMT Practical Soldering Lab	\$22.95
--------	-----------------------------	---------

AMFM108K	AM/FM IC Lab Kit & Course	\$36.95
----------	---------------------------	---------

AM2	AM Radio Learning Lab	\$11.95
-----	-----------------------	---------

AMF7	AM/FM Radio Learning Lab	\$12.95
------	--------------------------	---------

SR3	Short Wave Radio Learning Lab	\$16.95
-----	-------------------------------	---------

AK870	RC Speedster Car Kit	\$29.95
-------	----------------------	---------

Follow Us and SAVE \$\$

Follow us on your favorite network site and look for a lot of super deals posted frequently... exclusively for our followers!



Stereo Audio Platform Gain Controller

- ✓ Stereo audio processing while preserving audio dynamics!
- ✓ True stereo control keeps virtual sonic source location intact!
- ✓ Auto-bypass restores original levels when power is turned off!
- ✓ Built-in bar graph indication of signal level with display mute!



The SGC1 is one of our latest kits, and provides a great solution to the age-old problem: how can we easily correct inconsistent audio levels without negatively affecting the dynamics of the audio signal? The SGC1 circuit implements a principle known as the "Platform Gain Principle," which was originally developed by CBS Labs to allow transmitted audio levels to be automatically adjusted to keep them within a desired range.

Think of it like an audio engineer, constantly adjusting the output level in order to limit highs that would be too loud while boosting lower levels so that they can still be heard. You may think "oh, this is just another limiter/compressor!" Not so! Here's the real trick: keeping the full dynamic range ratio of the output signal the same as the original input - something the typical limiter/compressor can only dream of doing! The SGC1 can be placed in just about any standard analog stereo line audio circuit to keep the audio level within the desired range. It's also the perfect addition to any of our hobby kit transmitters, allowing you to match levels between different audio sources while keeping lows audible and preventing the highs from overdriving.

In addition to its useful basic function and great audio performance, the SGC1 also boasts a front panel LED meter to give an indication of the relative level of the input signal, as well as a front level control that allows you to adjust the controller to the min/max center point of your desired level range. And yes, it is a **Stereo Gain Controller!** Meaning that the levels of both the left and right channels are monitored and adjusted equally, thereby maintaining the relative virtual position of things like instruments, singers and speakers! The entire unit is housed in a slim attractive black textured aluminum case that is sure to complement your studio or home theatre. If you're looking for perfect audio levels, hire a broadcast audio engineer, but if that doesn't fit your budget, the SGC1 is the next best thing! Includes 15VDC world-wide power adapter.

SGC1 Stereo Audio Platform Gain Controller Kit

\$179.95

RF Preamplifier

The famous RF preamp that's been written up in the radio & electronics magazines! This super broadband preamp covers 100 KHz to 1000 MHz! Unconditionally stable gain is greater than 16dB while noise is less than 4dB! 50-75 ohm input. Runs on 12-15 VDC.

SA7 RF Preamp Kit

\$16.95



Mad Blaster Warble Alarm

If you need to simply get attention, the "Mad Blaster" is the answer, producing a LOUD ear shattering raucous racket! Super for car and home alarms as well. Drives any speaker. Runs on 9-12VDC.

MB1 Mad Blaster Warble Alarm Kit

\$9.95



Water Sensor Alarm

This little \$7 kit can really "bail you out"! Simply mount the alarm where you want to detect water level problems (sump pump)! When the water touches the contacts the alarm goes off! Sensor can even be remotely located. Runs on a standard 9V battery.

MK108 Water Sensor Alarm Kit

\$6.95



Air Blasting Ion Generator

Generates negative ions along with a hefty blast of fresh air, all without any noise! The steady state DC voltage generates 7.5kV DC negative at 400uA, and that's LOTS of ions! Includes 7 wind tubes for max air! Runs on 12-15VDC.

IG7 Ion Generator Kit

\$64.95



Tri-Field Meter Kit

THE GHOST DETECTOR!



"See" electrical, magnetic, and RF fields as a graphical LED display on the front panel! Use it to detect these fields in your house, find RF sources, you name it. Featured on CBS's Ghost Whisperer to detect the presence of ghosts! Req's 4 AAA batteries.

TFM3C Tri-Field Meter Kit

\$74.95

Electret Condenser Mic



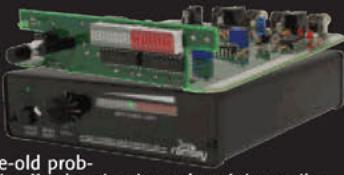
This extremely sensitive 3/8" mic has a built-in FET preamplifier! It's a great replacement mic, or a perfect answer to add a mic to your project. Powered by 3-15VDC, and we even include coupling cap and a current limiting resistor! Extremely popular!

MC1 Mini Electret Condenser Mic Kit

\$3.95

GET THE ~~NUTS AND VOLTS~~ DISCOUNT!

Mention or enter the coupon code **NVRMZ12** and receive 10% off your order!



The hit of the decade! Our patented receiver hears the entire aircraft band without any tuning! Passive design has no LO, therefore can be used on board aircraft! Perfect for airshows, hears the active traffic as it happens! Available kit or factory assembled.

ABM1 Passive Aircraft Receiver Kit

\$89.95



Measure RF with your standard DMM or VOM! This extremely sensitive RF detector probe connects to any voltmeter and allows you to measure RF from 100kHz to over 1GHz! So sensitive it can be used as a RF field strength meter!

RF1 Sniff-It RF Detector Probe Kit

\$27.95



IN-8-2 or IN-14 Tubes!

Classic Nixie Tube Clocks

Our next generation of classic Nixie tube clocks perfectly mesh today's technology with the Nixie era technology of the 60's. Of course, features you'd expect with a typical clock are all supported with the Nixie clock... and a whole lot more! The clocks are programmable for 12 or 24 hour mode, various AM/PM indications, programmable leading zero blanking, and include a programmable alarm with snooze as well as date display, 4 or 6 tube, kit or assembled!

We then jumped the technological time line of the 60's Nixie displays by adding the latest multi-colored LEDs to the base of the Nixie tubes to provide hundreds of illumination colors to highlight the glass tubes! The LED lighting can be programmed to any color and brightness combination of the colors red, green, or blue to suit your mood or environment. Then we leaped over the technological time line by integrating an optional GPS time base reference for the ultimate in clock accuracy! The small optional GPS receiver module is factory assembled and tested, and plugs directly into the back of the clock to give your Nixie clock accuracy you could only dream of! The clocks are available in our signature hand rubbed Maple, polished stainless, or clear acrylic bases.

NXIE Classic Nixie Tube Clock Kits From \$229.95

USB PIC Programmer



Finally, a compact USB PIC Programmer with a 20 pin ZIF socket for easy programming of most Microchip PIC Flash devices that does not require low voltage programming. Plus it uses USB therefore no more RS232 compatibility blues!

CK1301 USB PIC Programmer Kit

\$34.95

HV Plasma Generator



Generate 2" sparks to a handheld screwdriver! Light fluorescent tubes without wires! This plasma generator creates up to 25kV at 20kHz from a solid state circuit! Build plasma bulbs from regular bulbs and more! Runs on 16VAC or 5-24VDC.

PG13 HV Plasma Generator Kit

\$64.95

Speedy Speed Radar Gun



Our famous Speedy radar gun teaches you doppler effect the fun way! Digital readout displays in MPH, KPH, or FPS. You supply two coffee cans! Runs on 12VDC or our AC121 supply.

SG7 Speed Radar Gun Kit

\$74.95

Broadband RF Preamp



Need to "perk-up" your counter or other equipment to read weak signals? This preamp has low noise and yet provides 25dB gain from 1MHz to well over 1GHz. Output can reach 100mW! Runs on 12 volts AC or DC or the included 110VAC PS. Asmb.

PR2 Broadband RF Preamp

\$69.95

12VDC Regulated Supply



1A regulated supply. Worldwide input 100-240VAC with a Level-V efficiency! It gets even better, includes DUAL ferrite cores for RF and EMI suppression. All this at a 10 buck old wallwart price!

AC121 12VDC 1A Regulated Supply

\$9.95

800-446-2295
www.ramseykits.com

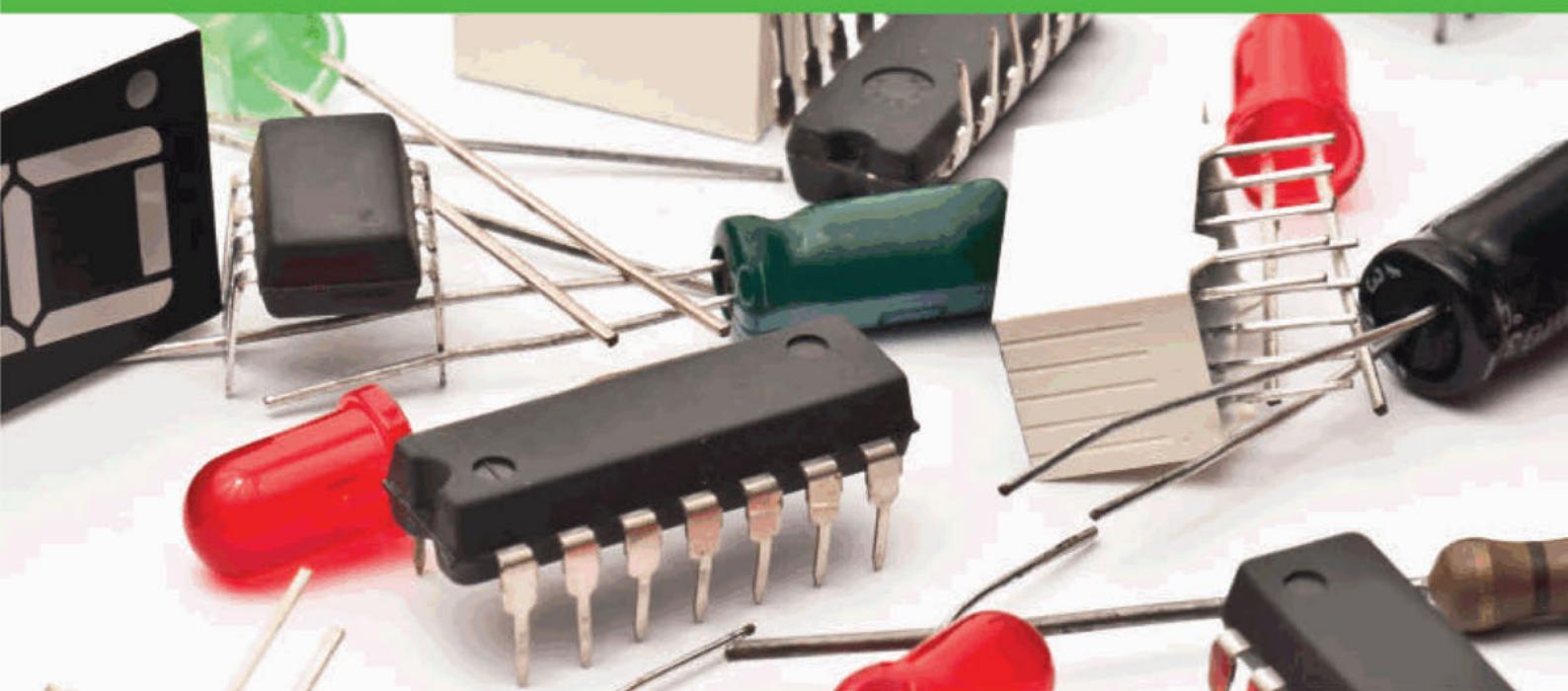
RAMSEY ELECTRONICS®

590 Fishers Station Drive
Victor, NY 14564
(800) 446-2295
(585) 924-4560

Prices, availability, and specifications are subject to change. Not responsible for typos, mistakes, stupidities, or New Year's morning headaches! Robin gets coal this year for making me get this ad copy done before the Turkey! Visit www.ramseykits.com for the latest pricing, specials, terms and conditions. Copyright 2013-2014 Ramsey Electronics® ... so there!

A BETTER PLACE FOR ELECTRONICS KITS

MORE KITS • WILDER KITS • YOUR KITS



- Over 600 electronics projects and kits for sale
- Electronic kits you won't find anywhere else
- All projects include components and instructions
- Design your own kit...We'll build it and pay you on every sale!

Join in the fun and visit us at:
www.ClubJameco.com/MoreKits

club
JAMECO®

